

Best Available Copy

(43)Date of publication of application : 23.04.1999

G06F 9/45

(72)Inventor : TANAKA AKIRA
KOTANI KENSUKE
TANAKA HIROHISA
SAYAMA JIYUNOKO

```

graph TD
    21[等値式信通生成制御部] --> 22[全呼式組合機圧部]
    21 --> 24[等値式組合生成部]
    22 --> 23[1式集合保持部]
    24 --> 25[E_FERN組合生成部]
    25 --> 26[E_CUT作次用1式集合保持部]
    25 --> 26_1[変数のE式判定部]
    26_1 --> 27[E_PRE組合生成部]
    27 --> 28[E_PRE組合保持部]
    27 --> 29[E_CUTWA組合保持部]
    29 --> 31[E_PRE組合保持部]
    31 --> 32[E_N組合保持部]
    32 --> 33[E_OUT組合保持部]
    33 --> 34[状態保持部]
    34 --> 35[DBE_OUT組合保持部]
  
```

<http://www19.ipdl.ncipi.go.jp/PA1/result/detail/main/wAAAyJaiHXDA41110225...> 2005/04/06

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-110225

(43) 公開日 平成11年(1999) 4月23日

(51) Int.Cl.⁶

識別記号

F I

G 0 6 F 9/45

G 0 6 F 9/44

3 2 2 F

審査請求 有 請求項の数24 O L (全 51 頁)

(21) 出願番号 特願平9-265655

(22) 出願日 平成9年(1997) 9月30日

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 田中 旭

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72) 発明者 小谷 謙介

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72) 発明者 田中 裕久

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(74) 代理人 弁理士 中島 司朗

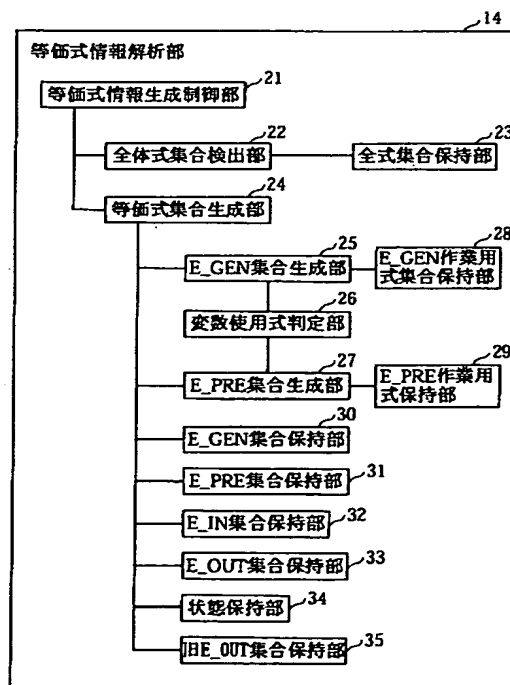
最終頁に続く

(54) 【発明の名称】 最適化装置及び最適化装置に適用されるプログラムが記録された記録媒体

(57) 【要約】

【課題】 高級言語で書かれたプログラムを機械語プログラムに翻訳するコンパイラに標準装備される最適化装置において、複数式間の等価性を短時間で解析する。

【解決手段】 等価性が成立する式からなる等価式集合の算出において、全ての基本ブロックに対して、当該基本ブロックの先行ブロックの出口点での等価式集合を使用して、当該基本ブロックの入口点での等価式集合を算出し、当該基本ブロックの入口点での等価式集合を使用して、当該基本ブロックの出口点での等価式集合を算出することを、全ての基本ブロックの出口点での等価式集合が変化しなくなるまで繰り返す。



【 特許請求の範囲】

【 請求項1 】 各命令の左辺及び右辺の何れか一方に位置する式を他の式に置き換えてもプログラムの実行が等価であることを示す式間の等価性を解析することにより、分岐元—分岐先の関係に基づいて複数の基本ブロックに分割されたプログラムを最適化する最適化装置であって、

分岐元の基本ブロックの出口において複数の式間に成立している等価性であって、分岐先の基本ブロックの入口まで維持されるものを各基本ブロックについて解析する第1 解析手段と、

分岐先基本ブロックの入口にまで維持された等価性が、基本ブロック内に位置する先頭命令から終了命令までの実行によりどれだけ消滅するかを解析し、それと共に先頭命令から終了命令までの実行により基本ブロックの内部においてどれだけの等価性が新たに成立するかを各基本ブロックについて解析する第2 解析手段と、

第2 解析手段による解析が行われると、第1 解析手段による解析により基本ブロック入口において等価性成立が明らかになった式からなる集合である等価式集合 E_IN と、第2 解析手段による解析により基本ブロック出口において等価性成立が明らかになった式からなる集合である等価式集合 E_OUT とがその基本ブロックの最適化に用いられるだけの適格性を有するかを全ての基本ブロックについて判定する適格性判定手段と、

適格性を有しない場合、第1 解析手段及び第2 解析手段による解析工程を繰り返し行わせるよう第1 解析手段及び第2 解析手段を起動する繰返手段と、適格性を有する場合、第1 解析手段及び第2 解析手段の最後の起動により生成された等価式集合 E_IN を用いて基本ブロック内に位置する先頭命令から終了命令までを最適化するブロック内最適化手段とを備えることを特徴とする最適化装置。

【 請求項2 】 請求項1 記載の最適化装置において、各基本ブロックに含まれている先頭命令から最終命令までの実行により等価性が成立する全ての式からなる等価式集合 E_GEN を各基本ブロックについて生成する等価式集合 E_GEN 生成手段と、

プログラム全域において用いられている全ての式からなる全式集合のうち、各基本ブロックに含まれている先頭命令から最終命令までの実行により更新されないものからなる等価式集合 E_PRE を各基本ブロックについて生成する等価式集合 E_PRE 生成手段と、

初期ブロック以外の基本ブロックについて等価式集合 E_GEN と、等価式集合 E_PRE との等価和演算を行なってその演算結果を等価式集合 E_OUT の初期集合とし、初期ブロックについて生成された等価式集合 E_GEN を初期ブロックの等価式集合 E_OUT の初期集合とする初期化手段とを備え、

前記第1 解析手段は、

初期化手段により生成された等価式集合 E_OUT の初期集合における式であって、その等価性が分岐先の基本ブロックの入口まで到達するものを各基本ブロックについて解析し、

第2 解析手段は、

分岐先基本ブロックの入口にまで到達した等価性が、基本ブロック内に位置する先頭命令から終了命令までの実行によりどれだけ消滅するかを解析し、それと共に先頭命令から終了命令までの実行により基本ブロックの内部において等価性がどれだけ発生するかを各基本ブロックについて解析することを特徴とする最適化装置。

【 請求項3 】 請求項2 記載の最適化装置における第1 解析手段は、

等価式集合 E_OUT が生成されると、同一の基本ブロックを分岐先とする基本ブロック同士で等価式集合 E_OUT の等価和演算を行うことにより、等価式集合 E_IN を再度生成する第1 演算手段を備え、

第2 解析手段は、

第1 演算手段による演算により生成された等価式集合 E_IN と、等価式集合 E_GEN 生成手段により生成された等価式集合 E_GEN と、等価式集合 E_PRE 生成手段により生成された等価式集合 E_PRE とを以下の{ 数式1 } に適用して等価式集合 E_OUT を再度生成する第2 演算手段とを備えることを特徴とする最適化装置。

{ 数式1 }

$$E_OUT[B] = E_GEN[B] \cup e(E_IN[B] \cap e E_PRE[B])$$

B: プログラム内の各基本ブロック

$\cap e$: 等価積演算の演算子 $\cup e$: 等価和演算の演算子

【 請求項4 】 請求項3 記載の最適化装置における適格性判定手段は、

第2 算出手段による演算により得られた等価式集合 E_OUT が適格性を有するかを判定し、

第1 演算手段は、

等価式集合 E_OUT が不適格であると判定されれば、同一の基本ブロックを分岐先とする基本ブロック同士の等価式集合 E_OUT の等価和演算を行うことにより、等価式集合 E_IN を再度生成し、

第2 演算手段は、

第1 演算手段による演算により生成された等価式集合 E_IN と、等価式集合 E_GEN 生成手段により生成された等価式集合 E_GEN と、等価式集合 E_PRE 生成手段により生成された等価式集合 E_PRE とを前記{ 数式1 } に適用して等価式集合 E_OUT を再度生成することを特徴とする最適化装置。

【 請求項5 】 請求項4 記載の最適化装置における適格性判定手段は、

第2 算出手段による演算により得られた等価式集合 E_OUT を過渡的な集合として記憶する過渡集合記憶部と、過渡集合を記憶した後、新たな等価式集合 E_OUT が第2 算出手段による演算により得られると、両者を照合する

10

20

30

40

50

照合部と、
照合した結果、記憶されている過渡集合と、新たな等価式集合 E_OUT とが一致すると、記憶された過渡集合が解としての適格性を満たすとの判定を下し、
照合した結果、記憶されている過渡集合と、新たな等価式集合 E_OUT とが不一致であると、記憶された過渡集合が解として不適格であるとの判定を下す判定部とを備えることを特徴とする最適化装置。

【請求項6】 請求項1～5の何れかに記載の最適化装置において、等価式集合 E_GEN 生成手段は、
等価性成立を解析すべき命令を取り出す第1取出部と、
各基本ブロックに含まれている先頭命令からそれまで取り出された命令までの間に等価性成立が明らかになった式の集合を保持している E_GEN 作業用保持部と、
取り出された命令が代入であり、かつ代入の左辺が変数である場合、当該変数の代入により影響を受ける式を前記 E_GEN 作業用保持部が保持する式集合から削除する第1の E_GEN 作業用保持部更新部と、
取り出された命令が代入であり、かつ代入の左辺が間接演算式である場合、当該間接演算式により更新される可能性のある変数によって影響を受ける式を前記 E_GEN 作業用保持部が保持する式集合から削除する第2の E_GEN 作業用保持部更新部と、
取り出された命令が代入であり前記 E_GEN 作業用保持部が保持する式集合の中に、当該命令における両辺のいずれかの式を含む式集合がある場合、当該命令の両辺の式を当該式集合に追加し、当該式集合が存在しないとき新たに当該命令の両辺を要素とする式集合を生成して前記 E_GEN 作業用保持部に追加する第3の E_GEN 作業用保持部更新部と、
命令が関数呼び出し命令である場合、関数呼び出し命令によって更新される可能性のある変数によって影響を受ける式を前記 E_GEN 作業用保持部が保持する式集合から削除する第4の E_GEN 作業用保持部更新部とを備えることを特徴とする最適化装置。

【請求項7】 請求項1～6の何れかに記載の最適化装置において、等価式集合 E_PRE 生成手段は、
等価性成立を解析すべき命令を取り出す第2取出部と、
プログラム全域にて存在する全ての式を保持する E_PRE 作業用保持部と、
取り出された命令が代入でありかつ代入の左辺が変数である場合、当該変数の代入により影響を受ける式を前記 E_PRE 作業用保持部から削除する第1の E_PRE 作業用保持部更新部と、
取り出された命令が代入でありかつ代入の左辺が間接演算式である場合、当該間接演算式により更新される可能性のある変数によって影響を受ける式を、前記 E_PRE 作業用保持部から削除する第2の E_PRE 作業用保持部更新部と、
取り出された命令が関数呼び出し命令である場合、関数

呼び出し命令によって更新される可能性のある変数によって影響を受ける式を、前記 E_PRE 作業用保持部から削除する第3の E_PRE 作業用保持部更新部とを備えることを特徴とする最適化装置。

【請求項8】 請求項3～7の何れかに記載の最適化装置において等価積演算は、
共通項を持たない互いに素な集合を要素とする二つの集合に対して、要素間の集合積演算を行なうことを特徴とする最適化装置。

10 【請求項9】 請求項3～8の何れかに記載の最適化装置において等価和演算は、
集合を要素とする二つの集合に対して、まず二つの集合の集合和演算を行ない、当該演算結果の中で、共通項をもつ集合同士を連結して一つの集合にするという操作を、それ以上連結できなくなるまで繰り返すことを特徴とする最適化装置。

【請求項10】 請求項1～9の何れかに記載の最適化装置におけるブロック内最適化手段は、
初期状態において前記第1算出手段により最後に算出された等価式集合 E_IN を保持し、同基本ブロック内の命令に対しての最適化が開始されると、その保持集合に対してこれまでに行われた同基本ブロック内の最適化に伴った更新がなされた等価式集合を保持している作業用保持部と、
基本ブロックの先頭命令から末尾命令までを順番に取り出す第3取出部と、
取り出された式に対して最適化を行う命令最適化手段と、
式が最適化されると、作業用保持部の保持集合を更新する更新手段とを備えることを特徴とする最適化装置。

【請求項11】 請求項2～10までの何れかに記載の最適化装置においてブロック内最適化手段は、
等価式集合の集合を保持する作業用保持部を前記等価式集合 E_IN に設定する作業用保持部初期化手段を備え、
命令最適化手段は、
取り出された命令を、等価式集合保持部が保持している等価式集合内の式のうち、その式を等価性を有する式に置換することにより冗長性を削除する第1の冗長性削除部と、

40 取り出された命令が代入でありかつ代入の両辺の式が、前記作業用保持部が保持する同一の式集合に含まれている場合、プログラム上の当該命令を削除する第2の冗長性削除部とを備え、
更新手段は、
取り出された命令が代入でありかつ代入の左辺が変数である場合、当該変数の代入により影響を受ける式を前記作業用保持部が保持する式集合から削除する第1の作業用保持部更新部と、
取り出された命令が代入でありかつ代入の左辺が間接演算式である場合、当該間接演算式により更新される可能

5

性のある変数によって影響を受ける式を、前記作業用保持部が保持する式集合から削除する第2の作業用保持部更新部と、

取り出された命令が代入でありかつ、前記作業用保持部が保持する式集合の中に、当該命令における両辺のいずれかの式を含む式集合がある場合、当該命令の両辺の式を当該式集合に追加し、当該式集合が存在しないとき新たに当該命令の両辺を要素とする式集合を生成し前記作業用保持部に追加する第3の作業用保持部更新部と、

取り出された命令が関数呼び出し命令である場合、関数呼び出し命令によって更新される可能性のある変数によって影響を受ける式を、前記作業用保持部が保持する式集合から削除する第4の作業用保持部更新部と、

前記作業用保持手段初期化手段を起動し、次に基本ブロック内の全ての命令に対して、前記第1、第2の冗長性削除部を起動しさらに、前記第1、2、3、4の作業用保持手段更新部を順番に起動する等価置換最適化制御部とを備えることを特徴とする最適化装置。

【請求項1 2】 請求項1 1記載の最適化装置においてブロック内最適化手段は、

命令が代入である場合、右辺の式が前記作業用保持部が保持する式集合の中に含まれていて、当該式集合に定数が含まれているときは、当該定数で当該命令の右辺を置き換え、

当該右辺の式が2項演算や単項演算であって当該演算で使用されている一つないし二つの変数が前記作業用保持部が保持する式集合の中に含まれており、かつ当該式集合に定数が含まれている場合、変数を定数に置き換えて当該演算を行ない、当該演算結果の定数値で当該命令の右辺を置き換える第3の冗長性削除部を有し、

前記等価置換最適化制御部は、

前記作業用保持部初期化手段を起動し、次に基本ブロック内の全ての命令に対して、前記第1、2、3の冗長性削除部を起動しさらに前記第1、2、3、4の作業用保持部更新部を順番に起動することを備えることを特徴とする最適化装置。

【請求項1 3】 請求項1 2記載の最適化装置においてブロック内最適化手段は、

命令が条件分岐である場合、条件式の両辺の式を含む前記作業用保持部が保持する式集合から、条件式を恒真または恒偽とすることが可能である場合、条件式を恒真を示す「(1)」または恒偽を示す「(0)」に置き換える第4の冗長性削除部を有し、

前記等価置換最適化制御部は、

前記作業用保持部初期化手段を起動し、次に基本ブロック内の全ての命令に対して、前記第1、2、3、4の冗長性削除部を起動しさらに、前記第1、2、3、4の作業用保持部更新部を順番に起動することを備えることを特徴とする最適化装置。

【請求項1 4】 各命令の左辺及び右辺の何れか一方に

6

位置する式を他の式に置き換えてもプログラムの実行が等価であることを示す式間の等価性を解析することにより、分岐元一分岐先の関係に基づいて複数の基本ブロックに分割されたプログラムを最適化する最適化装置に適用されるプログラムを記録した記録媒体であって、

前記プログラムは、

分岐元の基本ブロックの出口において複数の式間に成立している等価性であって、分岐先の基本ブロックの入口まで維持されるものを各基本ブロックについて解析する第1解析ステップと、

分岐先基本ブロックの入口にまで維持された等価性が、基本ブロック内に位置する先頭命令から終了命令までの実行によりどれだけ消滅するかを解析し、それと共に先頭命令から終了命令までの実行により基本ブロックの内サブステップにおいてどれだけの等価性が新たに成立するかを各基本ブロックについて解析する第2解析ステップと、

第2解析ステップによる解析が行われると、第1解析ステップによる解析により基本ブロック入口において等価性成立が明らかになった式からなる集合である等価式集合E_INと、第2解析ステップによる解析により基本ブロック出口において等価性成立が明らかになった式からなる集合である等価式集合E_OUTとがその基本ブロックの最適化に用いられるだけの適格性を有するかを全ての基本ブロックについて判定する適格性判定ステップと、適格性を有しない場合、第1解析ステップ及び第2解析ステップによる解析工程を繰り返し行わせるよう第1解析ステップ及び第2解析ステップを起動する繰返ステップと、

適格性を有する場合、第1解析ステップ及び第2解析ステップの最後の起動により生成された等価式集合E_INを用いて基本ブロック内に位置する先頭命令から終了命令までを最適化するブロック内最適化ステップとからなることを特徴とする記録媒体。

【請求項1 5】 請求項1 4記載の記録媒体に記録されたプログラムにおいて、各基本ブロックに含まれている先頭命令から最終命令までの実行により等価性が成立する全ての式からなる等価式集合E_GENを各基本ブロックについて生成する等価式集合E_GEN生成ステップと、プログラム全域において用いられている全ての式からなる全式集合のうち、各基本ブロックに含まれている先頭命令から最終命令までの実行により更新されないものからなる等価式集合E_PREを各基本ブロックについて生成する等価式集合E_PRE生成ステップと、初期ブロック以外の基本ブロックについて等価式集合E_GENと、等価式集合E_PREとの等価和演算を行なってその演算結果を等価式集合E_OUTの初期集合とし、初期ブロックについて生成された等価式集合E_GENを初期ブロックの等価式集合E_OUTの初期集合とする初期化ステップとを備え、

50

前記第1 解析ステップは、

初期化ステップにより生成された等価式集合 E_{OUT} の初期集合における式であって、その等価性が分岐先の基本ブロックの入口まで到達するものを各基本ブロックについて解析し、

第2 解析ステップは、

分岐先基本ブロックの入口にまで到達した等価性が、基本ブロック内に位置する先頭命令から終了命令までの実行によりどれだけ消滅するかを解析し、それと共に先頭命令から終了命令までの実行により基本ブロックの内サブステップにおいて等価性がどれだけ発生するかを各基本ブロックについて解析することを特徴とする記録媒体。

【請求項16】 請求項15記載の記録媒体に記録されたプログラムにおける第1 解析ステップは、等価式集合 E_{OUT} が生成されると、同一の基本ブロックを分岐先とする基本ブロック同士で等価式集合 E_{OUT} の等価和演算を行うことにより、等価式集合 E_{IN} を再度生成する第1 演算ステップを備え、

第2 解析ステップは、

第1 演算ステップによる演算により生成された等価式集合 E_{IN} と、等価式集合 E_{GEN} 生成ステップにより生成された等価式集合 E_{GEN} と、等価式集合 E_{PRE} 生成ステップにより生成された等価式集合 E_{PRE} とを以下の{ 数式1 } に適用して等価式集合 E_{OUT} を再度生成する第2 演算ステップとからなることを特徴とする記録媒体。

{ 数式1 }

$$E_{OUT}[B] = E_{GEN}[B] \cup (E_{IN}[B] \cap E_{PRE}[B])$$

B: プログラム内の各基本ブロック

\cap : 等価積演算の演算子 \cup : 等価和演算の演算子

【請求項17】 請求項16記載の記録媒体に記録されたプログラムにおける適格性判定ステップは、

第2 算出ステップにより得られた等価式集合 E_{OUT} が適格性を有するかを判定し、

第1 演算ステップは、

等価式集合 E_{OUT} が不適格であると判定されれば、同一の基本ブロックを分岐先とする基本ブロック同士の等価式集合 E_{OUT} の等価和演算を行うことにより、等価式集合 E_{IN} を再度生成し、

第2 演算ステップは、

第1 演算ステップによる演算により生成された等価式集合 E_{IN} と、等価式集合 E_{GEN} 生成ステップにより生成された等価式集合 E_{GEN} と、等価式集合 E_{PRE} 生成ステップにより生成された等価式集合 E_{PRE} とを前記{ 数式1 } に適用して等価式集合 E_{OUT} を再度生成することを特徴とする記録媒体。

【請求項18】 最適化装置は第2 算出ステップにより得られた等価式集合 E_{OUT} を過渡的な集合として記憶する過渡集合記憶部を備え、

請求項17記載の記録媒体に記録されたプログラムにお

ける適格性判定ステップは、

過渡集合を書き込んだ後、新たな等価式集合 E_{OUT} が第2 算出ステップにより得られると、両者を照合する照合サブステップと、

照合した結果、書き込まれた過渡集合と、新たな等価式集合 E_{OUT} が一致すると、記憶された過渡集合が解としての適格性を満たすとの判定を下し、

照合した結果、書き込まれた過渡集合と、新たな等価式集合 E_{OUT} が不一致であると、記憶された過渡集合が解として不適格であるとの判定を下す判定サブステップとからなることを特徴とする記録媒体。

【請求項19】 最適化装置は、各基本ブロックに含まれている先頭命令からそれまで取り出された命令までの間に等価性成立が明らかになった式の集合を保持している E_{GEN} 作業用保持部を備え、

請求項14～18の何れかに記載の記録媒体に記録されたプログラムにおいて等価式集合 E_{GEN} 生成ステップは、

等価性成立を解析すべき命令を取り出す第1 取出サブステップと、

取り出された命令が代入であり、かつ代入の左辺が変数である場合、当該変数の代入により影響を受ける式を前記 E_{GEN} 作業用保持部が保持する式集合から削除する第1 の E_{GEN} 作業用保持部更新サブステップと、

取り出された命令が代入であり、かつ代入の左辺が間接演算式である場合、当該間接演算式により更新される可能性のある変数によって影響を受ける式を前記 E_{GEN} 作業用保持部が保持する式集合から削除する第2 の E_{GEN} 作業用保持部更新サブステップと、

取り出された命令が代入であり前記 E_{GEN} 作業用保持部が保持する式集合の中に、当該命令における両辺のいずれかの式を含む式集合がある場合、当該命令の両辺の式を当該式集合に追加し、当該式集合が存在しないとき新たに当該命令の両辺を要素とする式集合を生成して前記 E_{GEN} 作業用保持部に追加する第3 の E_{GEN} 作業用保持部更新サブステップと、

命令が関数呼び出し命令である場合、関数呼び出し命令によって更新される可能性のある変数によって影響を受ける式を前記 E_{GEN} 作業用保持部が保持する式集合から削除する第4 の E_{GEN} 作業用保持部更新サブステップとからなることを特徴とする記録媒体。

【請求項20】 最適化装置はプログラム全域にて存在する全ての式を保持する E_{PRE} 作業用保持部を備え、請求項14～19の何れかに記載の記録媒体に記録されたプログラムにおいて等価式集合 E_{PRE} 生成ステップは、

等価性成立を解析すべき命令を取り出す第2 取出サブステップと、

取り出された命令が代入でありかつ代入の左辺が変数である場合、当該変数の代入により影響を受ける式を前記

E_PRE作業用保持部から削除する第1のE_PRE作業用保持部更新サブステップと、

取り出された命令が代入でありかつ代入の左辺が間接演算式である場合、当該間接演算式により更新される可能性のある変数によって影響を受ける式を、前記E_PRE作業用保持部から削除する第2のE_PRE作業用保持部更新サブステップと、

取り出された命令が関数呼び出し命令である場合、関数呼び出し命令によって更新される可能性のある変数によって影響を受ける式を、前記E_PRE作業用保持部から削除する第3のE_PRE作業用保持部更新サブステップとからなることを特徴とする記録媒体。

【請求項21】 最適化装置は、

初期状態において前記第1算出ステップにより最後に算出された等価式集合E_INを保持し、同基本ブロック内の命令に対しての最適化が開始されると、その保持集合に対してこれまでに行われた同基本ブロック内の最適化に伴った更新がなされた等価式集合を保持している作業用保持部を備え、

請求項14～20の何れかに記載の記録媒体に記録されたプログラムにおけるブロック内最適化ステップは、基本ブロックの先頭命令から末尾命令までを順番に取り出す第3取出サブステップと、

取り出された式に対して最適化を行う命令最適化ステップと、

式が最適化されると、作業用保持部の保持集合を更新する更新ステップとからなることを特徴とする記録媒体。

【請求項22】 請求項15～21までの何れかに記載された記録媒体におけるブロック内最適化ステップは、等価式集合の集合を保持する作業用保持部を前記等価式集合E_INに設定する作業用保持部初期化ステップを備え、

命令最適化ステップは、

取り出された命令を、等価式集合保持部が保持している等価式集合内の式のうち、その式を等価性を有する式に置換することにより冗長性を削除する第1の冗長性削除サブステップと、

取り出された命令が代入でありかつ代入の両辺の式が、前記作業用保持部が保持する同一の式集合に含まれている場合、プログラム上の当該命令を削除する第2の冗長性削除サブステップとを備え、

更新ステップは、

取り出された命令が代入でありかつ代入の左辺が変数である場合、当該変数の代入により影響を受ける式を前記作業用保持部が保持する式集合から削除する第1の作業用保持部更新サブステップと、

取り出された命令が代入でありかつ代入の左辺が間接演算式である場合、当該間接演算式により更新される可能性のある変数によって影響を受ける式を、前記作業用保持部が保持する式集合から削除する第2の作業用保持部

更新サブステップと、

取り出された命令が代入でありかつ、前記作業用保持部が保持する式集合の中に、当該命令における両辺のいずれかの式を含む式集合がある場合、当該命令の両辺の式を当該式集合に追加し、当該式集合が存在しないとき新たに当該命令の両辺を要素とする式集合を生成し前記作業用保持部に追加する第3の作業用保持部更新サブステップと、

取り出された命令が関数呼び出し命令である場合、関数呼び出し命令によって更新される可能性のある変数によって影響を受ける式を、前記作業用保持部が保持する式集合から削除する第4の作業用保持部更新サブステップと、

前記作業用保持ステップ初期化ステップを起動し、次に基本ブロック内の全ての命令に対して、前記第1、第2の冗長性削除サブステップを起動しさらに、前記第1、2、3、4の作業用保持ステップ更新サブステップを順番に起動する等価置換最適化制御サブステップとからなることを特徴とする記録媒体。

【請求項23】 請求項22記載の最適化装置においてブロック内最適化ステップは、

命令が代入である場合、右辺の式が前記作業用保持部が保持する式集合の中に含まれていて、当該式集合に定数が含まれているときは、当該定数で当該命令の右辺を置き換え、

当該右辺の式が2項演算や単項演算であって当該演算で使用されている一つないし二つの変数が前記作業用保持部が保持する式集合の中に含まれており、かつ当該式集合に定数が含まれている場合、変数を定数に置き換えて当該演算を行ない、当該演算結果の定数値で当該命令の右辺を置き換える第3の冗長性削除サブステップを有し、

前記等価置換最適化制御サブステップは、

前記作業用保持部初期化ステップを起動し、次に基本ブロック内の全ての命令に対して、前記第1、2、3の冗長性削除サブステップを起動しさらに前記第1、2、3、4の作業用保持部更新サブステップを順番に起動することを特徴とする記録媒体。

【請求項24】 請求項22記載の最適化装置においてブロック内最適化ステップは、

命令が条件分岐である場合、条件式の両辺の式を含む前記作業用保持部が保持する式集合から、条件式を恒真または恒偽とすることが可能である場合、条件式を恒真を示す「(1)」または恒偽を示す「(0)」に置き換える第4の冗長性削除サブステップを有し、

前記等価置換最適化制御サブステップは、

前記作業用保持部初期化ステップを起動し、次に基本ブロック内の全ての命令に対して、前記第1、2、3、4の冗長性削除サブステップを起動しさらに、前記第1、2、3、4の作業用保持部更新サブステップを順番に起

動することを特徴とする記録媒体。

【 発明の詳細な説明】

【 0 0 0 1】

【 発明の属する技術分野】本発明は、コンパイラに装備される最適化装置及び最適化装置に適用されるプログラムを記録した記録媒体に関する。

【 0 0 0 2】

【 従来の技術】近年、エレクトロニクス業界の技術者が最も苦心しているのは、より高度で複雑な制御を実現する組込型マイコンシステムの開発である。一般に組込型マイコンシステムとは、ファームウェアからアプリケーションプログラムまでの制御用プログラム全般をマスクROMに記録し、これと組み込み用途向けのマイクロプロセッサとを一体型に構成したコンピュータシステムをいい、その応用製品は家電製品、工作機械、情報機器、通信機器等さまざまな分野に活躍の場を広げている。

【 0 0 0 3】このようなマイコンシステムにおける組み込みプログラムは、C言語などの高級プログラミング言語で開発されることが一般的になっている。これは、組み込み用途の応用ソフトウェアの規模が著しく増大する傾向にあり、旧態依然としたアセンブリ言語を中心としたソフトウェア開発環境では、もはや組込型プログラムに要求される高度な処理を実現することが不可能だからである。また、高度な処理を実現する組込型プログラムをアセンブリ言語を用いて開発しようとすると技術者に多大な負担を強要してしまうからである。

【 0 0 0 4】とはいえ、アセンブリ言語により開発された応用ソフトウェアと比較して、高級プログラミング言語を用いて開発された機械語プログラムは冗長性が高いという欠点があり、より低価格な製品を開発しようとしているメーカー、高級プログラミング言語を用いた組み込み用途プログラムの開発を冷ややかな目で見るとある。

【 0 0 0 5】何故なら、組み込みプログラムは製品時に読みだし専用メモリ（ROM）に格納されるので、そのサイズの増大は製品コストに大きな影響を及ぼすからである。また特定の性能（実行速度）が要求される製品では、組み込みプログラムの実行時間の増大は、よりコストの高いマイクロプロセッサや、より高いクロック数におけるマイクロプロセッサの動作を要求してしまうからである。

【 0 0 0 6】現状では、これらの欠点が顕著な故に組み込み用途の分野における認知度は必ずしも高いとはいえない。組み込み用途の分野において高級プログラミング言語を用いたソフトウェア開発環境を主流とするには、このような冗長性を徹底して排除する高度な最適化アルゴリズムを確立する必要があり、研究者の手腕が問われるところである。

【 0 0 0 7】尚『プログラムの冗長性』には様々な解釈があるが、以降の説明における『プログラムの冗長性』

は、高級言語或は中間言語にて記述されたプログラムにおいて、最終的に得られる機械語プログラムのコードサイズや実行時間を増大させる全ての要因を意味することとする。従来の最適化装置について述べる前に、コンパイラの構成について説明する。説明に先立ち、本発明で引用する文献を上げる。

[1] A.V.Aho, R.Sethi, J.D.Ullman: "Compilers, Principle, Techniques, and Tool", Addison Wesley Publishing Company Inc., 1986

（邦訳）原田賢一： "コンパイラ I、II", サイエンス社, 1990

[2] Hans Zaima: "Supercompilers for Parallel and Vector Computers", Addison Wesley Publishing Company Inc., 1991

（邦訳）村岡洋一： "スーパーコンパイラ", オーム社, 1995

[3] 佐々 政孝: "プログラミング言語処理系", 岩波書店, 1989

図3 6 に従来のコンパイラの構成図を示す。コンパイラは構文解析装置4 1 と最適化装置4 2、コード生成装置4 9 とで構成されている。

【 0 0 0 8】構文解析装置4 1 は記憶装置（図外）にファイルとして記憶されているソースプログラムの字句解析、構文解析および意味解析を行ない、中間プログラムに変換する。ここで中間プログラムとは、コンパイラがコンパイラ内部でソースプログラムを処理しやすくするために変換した中間的なプログラムのことであり、中間プログラムの1 ステップを中間命令と呼ぶことにする。中間命令には文献 [1] p464 に示されているように、4 つ組や、3 つ組や、抽象式ツリーなどが存在し、これらが変換されて最終的なオブジェクトコードが生成される。尚、構文解析装置4 1 については、以降の主題である最適化との関連は希薄であるのでこれ以上の説明は省略する。

【 0 0 0 9】最適化装置4 2 は最終的に生成される機械語プログラムのプログラムサイズおよび処理実行時間を減少させる目的で中間プログラムの最適化を行なう。最適化装置4 2 は、最適化制御部4 3、制御フロー情報解析部4 4、データフロー情報解析部4 5、中間コード最適化部4 6、制御フロー情報保持部4 7、データフロー情報保持部4 8 より構成される。

【 0 0 1 0】制御フロー情報解析部4 4 は、制御の流れが一方向である基本ブロックに中間プログラムを分割し、基本ブロック間の制御の流れを示す、制御フロー情報を求める（文献 [1] p528）。基本ブロックの詳細な説明は後述の従来の最適化装置4 2 の説明で述べる。データフロー情報解析部4 5 は、文献 [1] p608 で示されているように、前述の制御フロー情報を使用して、さらに中間コードを解析し、到達する定義、利用可能式、生きている変数などのデータフロー情報を求める。詳細は文献

[1]p608-722に記載されている通りであり、概要は、まずいかなる情報をデータフロー情報とするかというデータフロー情報の設定、次に基本ブロックの入口出口でのデータフロー方程式の設定、そして繰り返しアルゴリズムによるデータフロー方程式の解を求めることによって、最終的なデータフロー情報を求めている。

【0011】中間コード最適化部46は、前述の制御フロー情報とデータフロー情報を使用して、中間コードの最適化を行なう。例えば、制御フロー情報を使用して制御が届かない基本ブロックを削除したり、前述の利用可能式の情報を使用した共通部分式の最適化(文献[1]p592)や、到達する定義の情報を使用したコピー伝搬の最適化(文献[1]p594)や、生きている変数の情報を使用した不要コード削除の最適化(文献[3]p482)など行なう。詳細については、コード生成装置の説明の後に述べる。

【0012】コード生成装置49は、中間プログラム中の変数にレジスタまたはメモリを割り付け、さらに中間命令毎に機械語命令に変換する。但し、このコード生成装置49は、本発明の主眼ではないのでこれ以上の説明は省略する。次に、従来の最適化装置42についてさらに詳しく説明する。但し、ここでは、本発明と関係のある、到達する定義の情報を使用した最適化であるコピー伝搬と、利用可能式の情報を使用した共通部分式の最適化について述べる。

【0013】まず、説明の前に以下で用いる用語について説明する。

・プログラムの点

ある中間命令と、その次に位置する中間命令との間を「点」と呼ぶ(文献[3]p461)。

・基本ブロック(文献[1]p528)

最適化のためにプログラムを書き換えるあたって、列の途中に飛び越し命令があったり列の途中が飛び越し先になっていれば命令の書き換えによってアルゴリズムが破壊される恐れがある。従って最適化にあたってはプログラムの実行順序が先頭から最後に向かうという一方向のみである必要がある。中間プログラムの一部分(ブロック)であり、そのブロックの途中からの飛び越しがなく、さらにブロックの途中への飛び越し先もないものを基本ブロックといい、最適化は、基本ブロックを最小単位にして行われる。基本ブロックの最初の中間命令の直前の点を基本ブロックの入口といい、基本ブロックの最後の中間命令の直後の点を基本ブロックの出口という。サブルーチン(関数)呼び出しの中間命令では基本ブロックは分割されないとする。何故なら、基本ブロックをより大きな領域と解釈する方が、より多くの最適化情報を解析できるからである。

【0014】以下の説明では、基本ブロックを中間命令の集合と考え、中間命令sが基本ブロックBに属していることを「 $s \in B$ 」と表す。また、文献[1]p532の示される

ように基本ブロックBの実行の前に実行される直前の基本ブロックをBの「先行ブロック」といい、基本ブロックBの先行ブロックの集合を $\text{pred}(B)$ と表す。同様に、基本ブロックBの直後に実行される基本ブロックをBの「後続ブロック」といい、基本ブロックBの後続ブロックの集合を $\text{succ}(B)$ と表す。基本ブロックの例を図3に示す。基本ブロックBLK2の先行ブロックは基本ブロックBLK1とBLK5であり、後続ブロックが基本ブロックBLK3とBLK4である。特に、最初に実行され、 $\text{pred}(B)$ を持たない基本ブロックは「初期ブロック」という。

・変数の定義、使用

文献[3]p.419に示すように変数に値を設定することを「定義する」といい、設定した値を使用することを「使用する」または「参照する」という。また、中間命令「 $s1: a = b \text{ op } c$ 」(a, b, c は変数、 op は演算子)において、中間命令s1は変数aの「定義」といい、また、中間命令s1は変数b, cの「使用」または「参照」という。以下の説明では簡単化のために、「 $s1(a=)$ 」で中間命令s1が変数aの「定義」であることを、「 $s1(=b)$ 」で中間命令s1が変数bの「使用」または「参照」であることを表す。

・到達する定義

これはデータフロー情報を論述する場合に、理解が求められる基本概念である。文献[3]p465-p466に示されているように、ある中間命令sにおいて変数xを使用したと仮定するとき、中間命令sにおいて変数xの値を設定した可能性のある定義dの集合のことを「中間命令sに到達する定義」という。

【0015】一方、ある変数xの定義dからある中間命令sに至る経路に変数xの定義d以外の定義が存在するとき『定義dは中間命令sに到達しない』という。定義d以外の定義が存在しないとき定義dは中間命令sに到達するという。そのような定義dの集合を中間命令sに到達する定義という。ここで必ずしも中間命令sにおいて変数xの使用があるとは限らないことに注意されたい。尚、これまでの説明で、中間命令sを「中間命令sの直前の点p」と読み換えてもよい。

【0016】ある基本ブロックに先行ブロックが複数あり、その複数の先行ブロック内の定義が基本ブロック内の中間命令に到達するという態様が存在する点も注意に値する。このような態様を図39の例で示す。変数b2の定義中間命令s17からs33に至る経路において、基本ブロックB1に位置し、変数b2を定義する定義中間命令s17は、中間命令s33に到達する定義である。同様に基本ブロックB2内に位置し、変数b2を定義する定義中間命令s29も中間命令s33に到達する定義である。つまり、変数b2には、2つの先行ブロック内に位置する複数の定義が一つの中間命令に到達する態様が存在する。さらに中間命令s33において変数t21の使用はないが、t21の定義中間命令s13からs33に至る経路にs13の他にt21の定義がない

とき、t21の定義中間命令s13もs33に到達する定義である。

・データフロー方程式

文献[3]p471-472で示されているように、『各中間命令にどのような定義が到達しているか』を示すデータフロー情報を算出するには、各データフロー情報毎にデータフロー方程式を設定し、繰り返し解法を用いてその方程式を解く必要がある。ここでは、到達する定義のデータフロー方程式について説明する。

【0017】文献[3]p471-472で示されているように、基本ブロックBにおいて $GEN[B]$ をBで生成される定義の集合とする、これは、基本ブロックBの中で定義され、その定義がBの出口まで到達する定義の集合である。形式的には次の{数式2}のようになる。

{数式2}

$GEN[B] = \{s \mid s(x) \in B, \text{ かつ } s \text{ から } B \text{ の出口までに } s'(x) \text{ なる } s' \text{ は存在しない。}\}$

次に、基本ブロックB内に定義中間命令 $s'(x)$ が存在する、基本ブロックB以外の基本ブロックに属するxの定義の集合を $KILL[B]$ とする。形式的には次の数式3のようになる。

{数式3}

$KILL[B] = \{s \mid s(x) \in (B \text{ 以外の任意の基本ブロック}), \text{ かつ } s'(x) \in B \text{ なる } s' \text{ が存在する。}\}$

次に、基本ブロックBの入口に到達する可能性のある定義の集合を $IN[B]$ 、基本ブロックBの出口に到達する可能性のある定義の集合を $OUT[B]$ とする。形式的には次の数式4のようになる。

{数式4}

$IN[B] = \{s \mid \text{定義中間命令 } s(x) \text{ において、} s \text{ から } B \text{ の入口までの経路の中に、} s'(x) \text{ なる } s' \text{ が存在しない。}\}$

$OUT[B] = \{s \mid \text{定義中間命令 } s(x) \text{ において、} s \text{ から } B \text{ の出口までの経路の中に、} s'(x) \text{ なる } s' \text{ が存在しない。}\}$

これらを用いると、データフロー方程式は次の数式5のように表される。

{数式5}

$IN[B] = \bigcup OUT[B'] \quad \dots (1)$

$B' \in \text{pred}(B)$

$OUT[B] = GEN[B] \cup (IN[B] - KILL[B]) \quad \dots (2)$

ここで、「 \cup 」は集合和、「 $-$ 」は集合の差を表す。数式5(1)は基本ブロックBの入口に到達する定義は、基本ブロックBの先行ブロックの出口に到達する定義の集合和であることを表し、数式5(2)は基本ブロックBの出口に到達する定義は、基本ブロックBで生成される定義 $GEN[B]$ と、基本ブロックBの入口に到達しかつBの中で定義されない変数の定義であることを表す。

【0018】数式5(1),(2)は、 $IN[B]$ 、 $OUT[B]$ を変数とみなした連立方程式になっており、この解を求めることにより、基本ブロックBに到達する $IN[B]$ の情報が求められる。 $IN[B]$ が求まれば、文献[3]p475に示されているよ

うに、基本ブロック内の各中間命令に達する定義は実行順に、 $IN[B]$ の内容を必要に応じて変更することにより容易に求まる。

・データフロー方程式を解く繰り返しアルゴリズム

文献[3]p473-p474に示されている到達する定義の場合のように、データフロー方程式を繰り返しアルゴリズムを使用して解くのが一般的である。到達する定義の繰り返しアルゴリズムは図37に示す通りである。

【0019】尚、図37において、repeat, for, ifの各文と、 $=$, $!=$, $=$ などの演算はC言語に準じ、false, trueは値0, 1であるとする。以下で述べる他のアルゴリズムの記述も同様である。上記のアルゴリズムは、全ての基本ブロックBの $OUT[B]$ の値が変化しなくなるまで $IN[B]$ と $OUT[B]$ を再計算し続け、変化しなくなった(つまり収束した)ときの $IN[B]$ 、 $OUT[B]$ の値が上記データフロー方程式の解となる。尚、繰り返しアルゴリズムの収束性に関しては、文献[2]p79-p88で示されているように、一般にデータフロー情報の集合が合流演算に関してセミラテスであること、例えば、到達する定義の場合では、到達する定義の集合が、先行ブロックの合流演算である集合演算「 \cup 」においてセミラテスであることを示す必要がある。

【0020】また各基本ブロックの影響を表す関数 f が単調関数であること、例えば、到達する定義の場合、基本ブロックBに関して、 $OUT[B]$ を求める式の $IN[B]$ を x とした、関数 $f(x) = GEN[B] \cup (x - KILL[B])$ が単調関数であることを示す必要がある。

・使用一定義連鎖情報、定義一使用連鎖情報(文献[3]p476)

到達する定義の情報から、さらに有用な情報として変数の各使用に到達する変数の定義をリストとして表すものを使用一定義連鎖情報という、図38において、変数b1の使用s32に到達するb1の定義がs12,s24であるとき、s32における変数b1の使用一定義連鎖情報は(s12,s24)というリストで表現される。

【0021】また、これとは逆に、定義が到達する使用をリストとして表すのが定義一使用連鎖情報である。図38においては、変数b1の定義中間命令s12はs32に到達する定義であるときs12における変数b1の定義一使用連鎖情報は(s32)というリストで表現される。s24についても同様である。また、図41において、変数x4の定義中間命令s5がs16,s27,s35に到達する定義であるときs5における変数x4の定義一使用連鎖情報は(s16,s27,s35)というリストで表現される。

【0022】尚、図38から図42において使用一定義連鎖情報、定義一使用連鎖情報は破線の矢印で図示しており、例えば図38ではs32からs12に向かう破線の矢印が変数b1の使用一定義を、s12からs32に向かう破線の矢印が変数b1の定義一使用連鎖情報を表している。

・利用可能式

これも、データフロー情報の一つである。文献 [3]p476 に示されているように、プログラムの入口からある中間命令sまでに至るすべての経路で式E: $x \text{ op } y$ (opは演算子) が評価され、その式Eの最後の評価から、sに至るまでに、 x や y の定義がないとき、式Eは中間命令sにおいて利用可能であるという。尚、これまでの説明で、中間命令sを「中間命令sの直前の点p」と読み換えてもよい。

【0023】以上を図4 1 の例で示す。中間命令s16から中間命令s35に至る経路で、s16が式「 $x4+y4$ 」の最後の評価であり（つまり、s16からs35の間にs16以外で式「 $x4+y4$ 」を実行する中間命令がない）、かつ変数 $x4$, $y4$ の定義が存在せず、さらに中間命令s27についてもs16と同様であるとき、中間命令s35において式「 $x4+y4$ 」は利用可能式である。

【0024】利用可能式に関するデータフロー情報も到達する定義と同様に、利用可能式に関するデータフロー方程式の設定と、繰り返しアルゴリズムにより計算される（文献[3]p476-p479）。次に上記の到達する定義および、利用可能式の情報を用いた最適化である、コピー伝搬による最適化と、共通部分式の最適化について述べる。

・コピー伝搬による最適化

これは、s: $x=y$ という形の中間命令（以下コピーという）があるとき、この x を使用する中間命令s'において、s'に到達する x の定義がsのみであるとき、s'における x を y に置き換え（この置き換えをコピー伝搬という）、さらに x の使用がs'の他に存在しないとき、sを削除する。例えば図4 0 において中間命令s6は中間命令s26, s30に到達する変数 $t32$ の唯一の定義であり、かつs26, s30のみがs6で定義される $t32$ の使用であるので、s26, s30における $t32$ を $x3$ に置き換え、s6の削除が可能となる（文献[3]p445）。

・共通部分式の最適化

これはすでに計算済みの式の評価（実行）を重複して評価しないようにする最適化である。

【0025】例えば図4 1 において、前述の利用可能式の説明で述べたように中間命令s35において式「 $x4+y4$ 」は利用可能式である。つまり中間命令s16, s27において式「 $x4+y4$ 」が評価済みであるので、s35によって再び式「 $x4+y4$ 」の評価を行なう必要がないということである。よって、共通部分式の最適化では新たに変数 w を導入して、図4 2 のように中間命令s16, s27, s35を置き換え、新たに中間命令s50, s51のコピーを挿入する。尚、図4 2 から推測されるようにこの最適化は、中間命令s50, s51のコピーを挿入しても、中間命令s35における式「 $x4+y4$ 」の評価を行なわないことで、コスト（プログラムサイズや実行時間）が削減できるときに行なわれる（文献[3]p446）。

【0026】

【発明が解決しようとする課題】しかしながら従来技術 50

では、データフロー情報を使用して、基本ブロックを跨いだ大域的な依存関係を用いた最適化を行おうとすると、コピー伝搬可能であることを裏付けるため、或は、命令の削除が可能であることを裏付けるための解析時間がかかり過ぎてしまうという問題点がある。これは、大域的な依存関係の解析を行う場合は、複数の基本ブロックが条件付き分岐により起動することにより、実行順序が何通りも存在し、これに併せて依存関係も何通りも存在することを念頭に置いた徹底した解析が求められるからである。また分岐元基本ブロックの制御フローが分岐先基本ブロックの制御フローに依存するという『フィードバック型の依存関係』を想定した解析が求められるからである。たとえ多大な解析時間を費やして大域的な依存関係の解析を行っても基本ブロックを跨いだ最適化が安全であるという確固たる根拠がないため、それを実行に移すことはやはり躊躇される。

【0027】最適化の安全性を裏付けるための解析が如何にして行われるかを図3 8 から図4 1 を参照しながら説明を行なう。図3 8 では中間命令s32において、変数 $a1$ と $b1$ は同じ値を保持しているの、単純に思考すると、s32は削除すべきように思える。しかしs32に到達する変数 $b1$ の定義が中間命令s12, s24と2つ存在し、唯一でない。唯一でないから、s12, s24はコピー伝搬の対象とすべきでない。よって、コピー伝搬とは異なる最適化を行なう必要がある。他の最適化を実行する場合でも、中間命令s32に到達する変数 $b1$ の定義がどのような中間命令にて行われ、その中間命令においてどのような使用がなされているかを調べておく必要がある。s12, s24がコピーであって、その右辺が変数 $a1$ であることを調べる必要がある。図3 8 の場合であれば、中間命令s32の変数 $b1$ の使用一定義の依存関係がまだ単純であり、変数 $b1$ の使用一定義連鎖情報を参照すれば最適化の要否を決定することができる。

【0028】続いて紹介する図3 9 の一例は、依存関係が複雑であり、使用一定義連鎖情報を詳しく解析する必要がある。図3 9 において、使用一定義連鎖情報、定義一使用連鎖情報から解るように、中間命令s33が削除可能であることは直感的に察知されるであろう。しかし中間命令s33が削除可能であると判断するのは余りにも軽率である。中間命令s33の削除に危険性がないことを裏付けるには、中間命令s33の変数 $b2$ の使用一定義連鎖情報に存在する変数 $b2$ の定義中間命令s17にまず着目し、その右辺の変数 $t21$ の使用一定義連鎖情報をさらに調べて、変数 $t21$ の定義中間命令s13に着目してその右辺を調べる必要がある。同様のことを中間命令s33の変数 $b2$ の使用一定義連鎖情報にあるもう一つの定義中間命令s29および、s29に到達する変数 $t22$ の定義中間命令s25についても調べる必要がある。このように使用一定義連鎖情報を次々に辿って、全てのケースについて変数 $b2$ が変数 $a2$ と同じ値を保持することを念入りに調べ上げること

より、中間命令s33が削除可能であるとの裏付けがなされる。

【0029】続いて紹介する図40の一例は、依存関係が更に複雑であり、使用-定義連鎖情報のみならず、定義-使用連鎖情報を詳しく解析する必要がある。つまり図40において中間命令s34が削除可能であることは直感的に理解できるが、中間命令s34の削除に危険が無いことを裏付けるには、使用-定義連鎖情報に加えて定義-使用連鎖情報を調べる必要がある。まず、図39のときと同様に中間命令s34の変数b3の使用-定義連鎖情報から中間命令s18,s15を辿ってa3とb3が同じ値を保持することを確認し、さらに、中間命令s30における変数t32の使用-定義連鎖情報から中間命令s6を検出し、さらにs6で定義される変数t32の定義-使用連鎖情報から中間命令s26を得てその左辺を調べて、初めて中間命令s34において、変数a3,b3が同じ値を保持することが解る。中間命令s26の右辺が変数a3でない場合は、さらに中間命令s6の他の定義-使用連鎖情報を辿る必要がある。

【0030】以上見てきたように、一般的に中間命令s:
x = yにおいて変数x, yが同じ値を保持するか否かを判定し、変数の書き換えが安全である旨を裏付けるには、使用-定義連鎖情報、定義-使用連鎖情報を辿って行く必要があるため、多大な解析時間が費される。次に図41についての最適化について述べる。図41において、中間命令s16,s20および中間命令s27,s31の使用-定義連鎖情報、定義-使用連鎖情報から解るように、中間命令s35の直前で変数a4は式「x4+y4」の値を保持しているので、s35は削除可能である。しかし前述の共通部分式の最適化では、中間命令s35はコピーに変わるものの、削除までは行なうことができなかった。よってこの最適化も前述の共通部分式による最適化とは異なる。さて、中間命令s35において変数a4が式「x4+y4」と同じ値を保持していることを検出するには、図38から図40で見てきたように、使用-定義連鎖情報、定義-使用連鎖情報の解析の他に、式「x4+y4」というような式の形まで含めて解析する必要がある、さらなる解析時間を要することとなる。

【0031】以上のような解析は、プログラム書き換えに伴う危険性を払拭するために必須となる処理であるが、余りにも多大な解析時間が費やされ、とてもではないが実用に耐えるものではない。また、この解析時間が有限時間内に収まるという保証がない。故に従来技術における最適化では、本質的に基本ブロックを跨いだ最適化に不充分であり、複数基本ブロックの間に多くの冗長性を残してしまうものと結論付けざるを得ない。

【0032】本発明の第1の目的は、より少ない解析時間でプログラム書き換えに伴う危険性を払拭することができ、より高度に冗長性を排除することができる最適化装置を提供することである。本発明の第2の目的は、基本ブロックを跨いだ大域的な最適化にも耐え得る高度な

解析能力を有する最適化装置を提供することである。

【0033】

【課題を解決するための手段】上記第1、第2の目的を解決するため、本明細書における最上位概念の発明では、各基本ブロックの出口において各式間に等価性がどのように発生し、発生した等価性が分岐先の各基本ブロックの入口にどのように分配されるかを解析するようにしている。分岐元基本ブロックの出口から分岐先基本ブロックの入口までの、丁度基本ブロック間の境界に相当する部位における各式間の等価性の解析は、従来から行われてきた解析技術の盲点であり、本明細書では、いわばこの盲点をつく形で大域的な依存関係の解析を実現している。

【0034】そこで本発明は、各命令の左辺及び右辺の何れか一方に位置する式を他の式に置き換えてもプログラムの実行が等価であることを示す式間の等価性を解析することにより、分岐元-分岐先の関係に基づいて複数の基本ブロックに分割されたプログラムを最適化する最適化装置であって、分岐元の基本ブロックの出口において複数の式間に成立している等価性であって、分岐先の基本ブロックの入口まで維持されるものを各基本ブロックについて解析する第1解析手段と、分岐先基本ブロックの入口にまで維持された等価性が、基本ブロック内に位置する先頭命令から終了命令までの実行によりどれだけ消滅するかを解析し、それと共に先頭命令から終了命令までの実行により基本ブロックの内部においてどれだけの等価性が新たに成立するかを各基本ブロックについて解析する第2解析手段と、第2解析手段による解析が行われると、第1解析手段による解析により基本ブロック入口において等価性成立が明らかになった式からなる集合である等価式集合E_{IN}と、第2解析手段による解析により基本ブロック出口において等価性成立が明らかになった式からなる集合である等価式集合E_{OUT}とがその基本ブロックの最適化に用いられるだけの適格性を有するかを全ての基本ブロックについて判定する適格性判定手段と、適格性を有しない場合、第1解析手段及び第2解析手段による解析工程を繰り返して行わせるよう第1解析手段及び第2解析手段を起動する繰返手段と、適格性を有する場合、第1解析手段及び第2解析手段の最後の起動により生成された等価式集合E_{IN}を用いて基本ブロック内に位置する先頭命令から終了命令までを最適化するブロック内最適化手段とを備えるように最適化装置を構成している。

【0035】『式間の等価性』とは、プログラム内のある式を他の式に置き換えてもプログラムの実行が等価であることを示すという性質をいい、本明細書では、基本ブロックの入口、出口において等価性が成立する全ての式を示すE_{IN}集合及びE_{OUT}集合を、基本ブロックの個別の解析により一意に得られるE_{GEN}集合、E_{PRE}集合から求めるためのデータフロー方程式を提示している。デ

ータフロー方程式の収束解が得られるまで、このデータフロー方程式を解く作業を繰り返せば、複数の基本ブロックが条件分岐により複雑に起動し、実行順序が何通りも存在する場合や分岐元基本ブロックの制御フローが分岐先基本ブロックの制御フローに依存するという『フィードバック型の依存関係』が存在する場合でも、基本ブロック間の境界に相当する部位の等価式集合を高精度に求めることができる。

【 0 0 3 6 】

【 発明の実施の形態】

(第1章 序論) 序論では、本実施形態の理解に必要な概念の定義、性質、用語について説明し、併せて新たに考案したデータフロー情報である「等価式集合」と、そのデータフロー方程式、およびその方程式を解く繰り返すアルゴリズム、および等価式集合を使った最適化について説明する。

(第1.1章 本実施形態にて用いるプログラム系の概念)

・ 中間命令

『 従来の技術』において述べた通り 中間命令には数種類存在するが、ここでは説明を明確化するために、もっとも一般的な3番地形式のものを取り扱う。尚、以下で述べる本方式は他の中間命令にも適用可能である。

【 0 0 3 7 】 中間命令の形式は次の通りである。但し、本発明の説明に必要な基本的な部分についてのみ述べる。

(1) 代入中間命令「 a = b 」 ... ここで、aは、変数が間接演算式であり、bは変数、定数、単項演算子または、2 項演算子の式である。なお演算子はC 言語に準ずるものとする。例えば図3 の中間命令 s1, s17がその例である。

【 0 0 3 8 】 (2) 条件分岐中間命令 ... この中間命令は、「 if (条件式) goto (飛び先ラベル) 」という形式をとる。条件式はC 言語に準ずるものとする。例えば図3 の中間命令 s8, s39がその例である。

(3) 無条件分岐中間命令 ... この中間命令は、「 goto (飛び先ラベル) 」という形式をとる。例えば図3 の中間命令 s24がその例である。

【 0 0 3 9 】 (4) 関数(サブルーチン) 呼び出し 中間命令 ... この中間命令は、「 関数名 (実引数リスト) 」という形式をとる。例えば図3 の中間命令 s20がその例である。

(5) ラベル ... この中間命令は、中間プログラムの特定の場所を表すものである。例えば図3 の L1, L2, L3 がその例である。

・ 式

1996年度版JISハンドブック 58『 情報処理 用語・符号・データコード 編』によれば、「式」は『 一つ以上のオペランドから値を計算する言語構成要素』と解説されている。JISハンドブックにおけるこの「式」の解説に準拠して、本明細書における「式」を、変数、定数、単項式、2 項式 50

を始めとする言語構成要素と定義する。特に単項式のうちC 言語のような間接演算子「 * 」を使用した「 * (変数) 」形式の式を間接演算式という (尚「 * (変数) 」形式で使用される変数をポインタ変数という。) 。

・ 指示変数集合

変数vがポインタ変数であるとき、中間命令 s1において変数vが指し示す可能性のある変数の集合を指示変数集合ということにし、この集合を POINT[s,v]と表現することとする。指示変数集合の算出方法は、文献 [1]p648-660 で述べられておりその詳細な説明は省略するが、図1 3 (a)を使ってその概念を簡単に説明する。図1 3 (a)において、まず中間命令 s1, s2の直後から中間命令 s3の直前に至る経路までに、ポインタ変数pへの定義がないとする。このとき、中間命令 s1, s2では、それぞれポインタ変数pに変数a, bのアドレスが代入されているので、中間命令 s3において、変数pは変数a, bを指すことが解る。よって、中間命令 s3における指示変数集合 POINT[s3, p]には変数a, bが含まれる。また、中間命令 s3では変数aかまたはbが変更され、言い換えると、中間命令 s3は変数a, bの定義であるといえる。同様に中間命令 s4は、変数a, bの使用があるといえる。

【 0 0 4 0 】 尚、この指示変数集合の算出は、本方式の等価式集合の算出をする前に既に図1 のデータフロー情報解析部1 3で行なっているものとする。

・ 変更変数集合

中間命令 s1における関数fの呼び出しにより、値が変化し可能性のある変数の集合を変更変数集合ということにし、この集合を CHANGE[s, f]と表現することとする。変更変数集合の算出方法は、文献 [1]p648-660で述べられておりその詳細な説明は省略するが、図1 3 (b)を使ってその概念を簡単に説明する。図1 3 (b)は中間プログラムにおける関数の呼び出し関係を、C 言語記述的に示しており、変数aは外部変数を示し、関数f1は局所変数として変数b1, b2を持つことを示している。さらに中間命令 s1, s2において変数b1, b2のアドレスを実引数として渡して、関数f2を呼び出している。また、関数f2では、矢印y1のように仮引数であるポインタ変数pを使用して、その指示先に値を代入しており、さらに、矢印y2のように外部変数aに値を設定している。

【 0 0 4 1 】 よって、関数f1における中間命令 s1の関数f2の呼び出しでは、変数b1, aの値が変更される可能性があり、集合 CHANGE[s1, f2]に変数b1, aが含まれることになる。同様に、集合 CHANGE[s2, f2]に変数b2, aが含まれることになる。尚、この変更変数集合の算出は、本方式の等価式集合の算出をする前に既に図1 のデータフロー情報解析部1 3で行なっているものとする。

・ 等価式集合

プログラムにおいて、ある式aを別な式bに置き換え可能となき、「式bは式aに等価である」又は「式bは式aと等価性を有する」ということにする。また等価である式同

士は、「等価な関係にある」ということにする。

【0042】等価な関係にある式を一つの集合にまとめたものを「等価式集合」ということにする。尚、本明細書において式は、変数、定数、単項式、2項式と定義されているので、例えば、図41において、中間命令s20の直後から中間命令s35の直前までの間で変数a4の定義が存在せず、さらに中間命令s31の直後から中間命令s35の直前までの間で変数a4の定義が存在しないとき、中間命令s35の直前で式「x4+y4」と「a4」はお互いに等価な式となり、等価式集合(x4+y4, a4)が得られる。そして、この等価式集合から式「x4+y4」が式「a4」に置き換え可能となり、結果、中間命令s35を削除できることが判明する。

【0043】一般的に、等価式集合はプログラムの点ごとに異なり、各点において等価式集合が複数個存在する。例えば、図41において先ほどの変数a4の仮定の他に、中間命令s5の直後から中間命令s35の直前まで至る全ての経路に、変数x4,p41の定義がなく、さらに中間命令s7の直後から中間命令s35の直前まで至る全ての経路に変数y4,p42の定義がないとき、中間命令s35の直前では{(x4,p41), (y4,p42), (a4,x4+y4)}と複数の等価式集合が得られる。また同じ仮定において中間命令s31の直後では{(x4,p41), (y4,p42), (t42,x4+y4,a4)}となる。つまり、一般にプログラムの点ごとに等価式集合の集合(以下、等価式集合群という場合もある)が存在する。

【0044】また、図41の中間命令s5において、変数*

{ 数式7 }

$$E = \{e_1, e_2, \dots, e_n\}$$

$$P(E) = \{(e_1), (e_2), \dots, (e_1, e_2), \dots, (e_1, e_2, e_3), \dots, E\}$$

$$P^2(E) = \{\{(e_1)\}, \{(e_2)\}, \dots$$

$$\{(e_1, e_2)\}, \dots$$

$$\{(e_1, e_2, e_3), (e_3, e_4)\}, \dots$$

$$\dots, \{(e_1, e_2), (e_3, e_4), (e_5, e_6, e_7)\}, \dots$$

$$P(E) \dots\}$$

$$P^4(E) = \{\{\{(e_1)\}, \{(e_2)\}\}, \dots$$

$$\{\{(e_1, e_2)\}, \{(e_1, e_2, e_3), (e_3, e_4)\}\}, \dots$$

$$\{\{(e_1, e_3), (e_3, e_5), (e_5, e_7, e_9)\}, \{(e_2, e_4, e_6), (e_4, e_8)\}\}, \dots$$

$$P(E)^{2\dots}\}$$

・ 集合PR(E)

$P^2(E)$ に属するもので、要素の集合が互いに素であるものの集合全体をPR(E)とする。形式的には次の数式8のようになる。

{ 数式8 }

$$PR(E) = \{S \mid S \in P^2(E), S_i \in S, S_j \in S, S_i \neq S_j, S_i \cap S_j = \phi\}$$

例えば、{(e1, e2), (e3, e4), (e5, e6, e7)}はPR(E)に属するが、{(e1, e2, e3), (e3, e4)}では、e3が両方の要素に含まれているのでPR(E)に属さない。

【0046】尚、後で示されるように等価集合群は、その性質からPR(E)に属する。

*x4と変数p41との等価関係が生成し、この関係が中間命令s35の直前でも壊れず成立していることであるので、このことを変数xと変数p1の等価な関係が、中間命令s35の直前に「到達する」ということにする。

(第1.2章 本実施形態で引用する数学的な記述および用語) 以下では、集合論における一般的な用法である記号を用いる。すなわち、集合の要素であることを「 \in 」で、集合の和を「 \cup 」、集合の共通部分を「 \cap 」、集合の差を「 $-$ 」、集合同士が等しいかまたは一方に含まれることを「 \subseteq 」で表現する。また空集合を「 ϕ 」と表現する。また共通部分が空集合である集合同士は「互いに素である」という。また、全称記号を「 \forall 」、限定記号を「 \exists 」で表現する。

・ 式の全体集合E

中間命令のプログラムに現れる条件式以外の式の全体集合をEとする。例えば、図3に記載されている式が全てであれば、式の全体集合Eは図5(a)となる。

【0045】また、全体集合Eは一般的に数式6に示すように表現される有限集合である。

{ 数式6 }

$$E = \{e_1, e_2, \dots, e_n\}$$

以下の説明を進めるにあたり、集合E、その巾集合(E の部分集合の全体) $P(E)$ 、 $P(E)$ の巾集合($P(E)$ の部分集合の全体) $P^2(E)$ とする。さらに、 $P^2(E)$ の巾集合($P(E)^2$ の部分集合の全体) $P^4(E)$ とする。集合Eとともに、一般的に次の数式7のように表現される。

・ \subset 関係

集合 $X \in P^2(E)$ 上での \subset 関係を次に示す。

(1) Xの要素A, Bに共通部分が存在するとき、A, Bは \subset 関係にあるという。形式的には次のようになる。

【0047】A, B \in X, $A \cap B \neq \phi$ であるとき、A, Bは \subset 関係にあるといい、これを「 $A \subset B$ 」と表現する。

(2) Xの要素A, B, Cにおいて、A, Bが \subset 関係にあり、B, Cも \subset 関係にあるときA, Cは \subset 関係にあるという。形式的には次のようになる。A, B, C \in X, A, Bは \subset 関係にありかつ、B, Cも \subset 関係にあるとき、A, Cは \subset 関係にある。つまり、 $A \subset B$ かつ $B \subset C$ ならば $A \subset C$ 。

(3) Xの要素Aは自身と \subset 関係にある。形式的には次のよ

うになる。

【 0 0 4 8 】 $A \in X$ 自身とは α 関係にある。つまり、 $A \subset m A$ 。例えば、 $X = \{(e1, e2), (e2, e3), (e3, e4), (e5, e6)\}$ において、それぞれの要素を、 A, B, C, D とすると、 A, B は共通部分 $e2$ 、 B, C は共通部分 $e3$ が存在するので、 $A \subset m B$ 、 $B \subset m C$ 、 $A \subset m C$ である。しかし、 D は A, B, C のどれとも共通部分が存在しないので、 A, B, C とは α 関係にならない。

【 0 0 4 9 】 また、 α 関係の定義から、 α 関係は文献 [2] p351にあるような反射的 (3)、対称的 (1) かつ推移的 (2) な関係である。よって α 関係は同値関係であり、集合 $X \in P^2(E)$ は α 関係によって、互いに素な部分集合に分割できる。例えば、 $X = \{(e1, e3), (e2, e4, e6), (e3, e5), (e4, e8), (e5, e7, e9)\}$ は、 α 関係によって $X1 = \{(e1, e3), (e3, e5), (e5, e7, e9)\}$ 、 $X2 = \{(e2, e4, e6), (e4, e8)\}$ という 2 つの部分集合に分割できる。

【 0 0 5 0 】 ここで注目する点は、分割の結果できた特定の部分集合に属する集合の要素、例えば上述の $X1$ における $e3$ は、他の部分集合に属する集合、例えば上述の $X2$ のどの要素の集合にも属さない。

・ 集合 $CM_PT[X]$

{ 数式 1 0 }

$E_SET[Y]$

$= E_SET[CM_PT[\{(e1, e3), (e2, e4, e6), (e3, e5), (e4, e8), (e5, e7, e9)\}]]$

$= E_SET[\{(e1, e3), (e3, e5), (e5, e7, e9)\}, \{(e2, e4, e6), (e4, e8)\}]$

$= \{(e1, e3, e5, e7, e9), (e2, e4, e6, e8)\}$ となる。

【 0 0 5 3 】 更に図 3 5 は、 $E_SET[CM_PT[X]]$ 集合を図式的に表現したものであり、図 3 5 (a) は次の数式 1 1 に示す集合のベン図である。

{ 数式 1 1 }

$X1 = (e1, e3)$

$X2 = (e2, e4, e6)$

$X3 = (e3, e5)$

$X4 = (e4, e8)$

$X5 = (e5, e7, e9)$

さらに、図 3 5 (b) は $CM_PT[X]$ 集合であり、図 3 5 (c) は $E_SET[CM_PT[X]]$ 集合を図式的に表現したものである。

このように、もともと共通部分が存在する部分で連結された集合 (例えば図 3 5 (a) の $X1, X3, X5$) が、 $E_SET[CM_PT[X]]$ 集合では一つの集合 (例えば図 3 5 (b) の $X1 \cup X3 \cup X5$) を形成することを示している。さらに、 $E_SET[CM_PT[X]]$

{ 数式 1 3 }

$A \cup e B = E_SET[CM_PT[A \cup B]]$

$= E_SET[CM_PT[\{(e1, e3), (e2, e4, e6), (e4, e8), (e5, e7, e9)\}]]$

$= E_SET[\{(e1, e3)\}, \{(e2, e4, e6), (e4, e8)\}, \{(e5, e7, e9)\}]$

$= \{(e1, e3), (e2, e4, e6, e8), (e5, e7, e9)\}$

また直観的には、前述の図 3 5 の説明からも解るように、集合 $E_SET[CM_PT[X]]$ 自体は、集合 $X \in P^2(E)$ の要素に対して、「共通部分をもつもの同士を連結して一つの集合とする」という操作を、共通部分をもつものがな

* 集合 $CM_PT[X]$ は、集合 $X \in P^2(E)$ を α 関係により分割した結果できる分割の集合である。尚、明らかに $CM_PT[X] \in P^4(E)$ である。

【 0 0 5 1 】 例えば、上述の、 $X = \{(e1, e3), (e2, e4, e6), (e3, e5), (e4, e8), (e5, e7, e9)\}$ において、 $CM_PT[X] = \{\{(e1, e3), (e3, e5), (e5, e7, e9)\}, \{(e2, e4, e6), (e4, e8)\}\}$ となる。

・ 集合 $E_SET[Y]$

集合 $Y \in P^4(E)$ に属する要素集合 $P \in P^2(E)$ 毎に、集合 P に属する集合 $E \in P(E)$ の要素 $e \in E$ で構成される、集合 $S \in P(E)$ を集めたものが $E_SET[Y] \in P^2(E)$ である。形式的には次の数式 9 のようになる。{ 数式 9 }

$Y \in P^4(E)$,

$E_SET[Y]$ 集合 $= \{S \mid \exists P \in Y, S = \bigcup E_j\}$

$E_j \in P_i$ である。

【 0 0 5 2 】 例えば、上述の、 $X = \{(e1, e3), (e2, e4, e6), (e3, e5), (e4, e8), (e5, e7, e9)\}$ および $Y = CM_PT[X] = \{\{(e1, e3), (e3, e5), (e5, e7, e9)\}, \{(e2, e4, e6), (e4, e8)\}\} \in P^4(E)$ の例で示すと、次の数式 1 0 のようになる。

20

*

※ $PT[X]$ 集合はその要素である、 $X1 \cup X3 \cup X5$ および、 $X2 \cup X4$ は互いに素であるので、 $E_SET[CM_PT[X]]$ 集合は、 $PR(E)$ 集合に属する集合である。なおこの性質は、 X 以外の一般の $P^2(E)$ に属する集合についても同様に成り立つことである。

30

【 0 0 5 4 】 ・ 等価和演算の定義 (演算子「 $\cup e$ 」の定義)

次の数式 1 2 に示されるような集合演算を等価和演算と呼び、その演算子を「 $\cup e$ 」で表現するとする。

{ 数式 1 2 }

$A, B \in PR(E)$

$A \cup e B = E_SET[CM_PT[A \cup B]]$

例えば、 $A = \{(e1, e3), (e2, e4, e6)\}$ 、 $B = \{(e1, e3), (e4, e8), (e5, e7, e9)\}$ のとき、 $A \cup e B$ は次の { 数式 1 3 } のようになる。

くなるまで繰り返した結果にできる集合に等しい。

・ 等価和演算子 $\cup e$ の性質

文献 [2] p81 の意味において、次のように $PR(E)$ 上の演算子 $\cup e$ について以下の性質がなり立つ。

50

(1) $A \cup_e A = A$ (同一性)

[証明] 和集合「 \cup 」の性質により、 $A \cup A = A$ であり、 $A \in PR(E)$ であるから

$A \cup_e A = E_SET[CM_PT[A \cup A]] = E_SET[CM_PT[A]] = A$ が成り立つ。(2) $A \cup_e B = B \cup_e A$ (可換性)

[証明] 和集合「 \cup 」の性質により、 $A \cup B = B \cup A$ であるので、

$A \cup_e B = E_SET[CM_PT[A \cup B]] = E_SET[CM_PT[B \cup A]] = B \cup_e A$

が成り立つ。

(3) $(A \cup_e B) \cup_e C = A \cup_e (B \cup_e C)$ (連合性)

[証明] 演算 \cup_e の定義より明らかに、まず $E_SET[CM_PT[A \cup B]] = X$ を行ない、さらに、 $E_SET[CM_PT[X \cup C]]$ を行なったものは、はじめから、 $E_SET[CM_PT[A \cup B \cup C]]$ を行なったものに等しい。よって、 $(A \cup_e B) \cup_e C = E_SET[CM_PT[A \cup B \cup C]]$ である。

【0055】同様に、 $A \cup_e (B \cup_e C) = E_SET[CM_PT[A \cup B \cup C]]$ でもあるので、 $(A \cup_e B) \cup_e C = A \cup_e (B \cup_e C)$ が成り立つ。

・等価積演算の定義 (演算子「 \cap_e 」の定義)

次の数式14に示されるような集合演算を等価積演算と呼び、その演算子を「 \cap_e 」で表現するとする。

{数式14}

$A, B \in PR(E)$

$A \cap_e B = \{C \mid A_i \in A, B_i \in B, C = A_i \cap B_i\}$

例えば、 $A = \{(e1, e3), (e2, e4, e6)\}$, $B = \{(e1, e3), (e4, e8), (e5, e7, e9)\}$ のとき、 $A \cap_e B = \{(e1, e3), (e4)\}$ *

{数式15}

$A, B, C \in PR(E)$ に対して、

$\forall w_i \in (A \cap_e C) \cup_e (B \cap_e C),$

$\exists x_j \in (A \cup_e B) \cap_e C,$

$w_i \subseteq x_j$

... (a)

(全ての $w_i \in (A \cap_e C) \cup_e (A \cap_e C)$ に対して、

$x_j \in (A \cup_e B) \cap_e C$ が存在し、 $w_i \subseteq x_j$ である。)

[証明] まず、 \cup_e の定義より、

(1) $\forall A_k \in A, \exists x_l \in A \cup_e B, A_k \subseteq x_l$ であり、

(2) $\forall B_m \in B, \exists x_n \in A \cup_e B, B_m \subseteq x_n$ である。

(3) (1)より、 $\forall C_p \in C$ について、

$A_k \cap C_p \subseteq x_l \cap C_p$

ここで、 $(A_k \cap C_p) \in (A \cap_e C)$, $(x_l \cap C_p) \in (A \cup_e B) \cap_e C$ であり、 A_k, C_p は任意であるから、

$\forall A_k \in (A \cap_e C), \exists y_j \in (A \cup_e B) \cap_e C, A_k \subseteq y_j$

である。

(4) (3)と同様に(2)より、 $\forall C_q \in C$ について、

$(B_m \cap C_q) \subseteq (x_n \cap C_q)$

ここで、 $(B_m \cap C_q) \in (B \cap_e C)$, $(x_n \cap C_q) \in (A \cup_e B) \cap_e C$ であり、 B_k, C_q は任意であるから、

$\forall B_k \in (B \cap_e C), \exists y_k \in (A \cup_e B) \cap_e C, B_k \subseteq y_k$

* となる。

・等価積演算子 \cap_e の性質

文献[3]p81の意味において、次のように集合 $PR(E)$ 上の演算子 \cap_e について以下の性質が成り立つ。

(1) $A \cap_e A = A$ (同一性)

[証明] $A_i \in A, A_j \in A, i = j$ のとき $A_i \cap A_i = A_i$ であり、 $i \neq j$ において、 $A_i \cap A_j = \phi$ であるので $A \cap_e A = A$ が成立する。例えば、 $A = (A1, A2, \dots, An)$ において、 $A1$ に着目すると、 $A1 \cap A_j = \phi$ ($n \geq j > 1$) であるので、結局 $A1 \cap A = A1$ しか残らない。

(2) $A \cap_e B = B \cap_e A$ (可換性)

[証明] \cap_e の定義の集合の共通部分「 \cap 」の性質より明らか。

(3) $(A \cap_e B) \cap_e C = A \cap_e (B \cap_e C)$ (連合性)

[証明] \cap_e の定義の集合の共通部分「 \cap 」の性質より明らか。

(4) 全ての $A \in PR(E)$ において、 $A \cap_e \phi = \phi$ であるので、ゼロ要素は ϕ である。

(5) 全ての $A \in PR(E)$ において、 $A \cap_e \{E\} = A$ であるので、1要素は $\{E\}$ である。

【0056】以上により、文献[3]p81の意味において、集合 $PR(E)$ は演算 \cap_e においてセミラテスである。

・演算子 \cup_e への演算子 \cap_e の分配

ここでは、後述の「関数 $f(X) = E_GEN[B] \cup_e (X \cap_e \{E_PRE[B]\})$ の単調性の証明」で使用する演算 \cup_e に演算 \cap_e を分配したときの次の数式15の命題について証明する。

29

数の要素をもつときは、任意の $E_1, E_n \in P_i$ について、
 $E_1 \cap E_2 \neq \phi$

.....

$E_{t-1} \cap E_t \neq \phi$

$E_t \cap E_{t+1} \neq \phi$

.....

$E_{n-1} \cap E_n \neq \phi$

なる、 $E_t \in P_i$ ($t=1..n$)が存在する。

(9) ここで、(5)より、 $E_t \subseteq Y_t$, $E_{t+1} \subseteq Y_{t+1}$ ($Y_t, Y_{t+1} \in ((A \cup e B) \cap e C)$)が存在し、かつ、 $E_t \cap E_{t+1} \neq \phi$ であるから、共通項 $Y_c = E_t \cap E_{t+1}$ は、 Y_t, Y_{t+1} 両方に含まれる。つまり、

$Y_c \subseteq Y_t$ かつ、 $Y_c \subseteq Y_{t+1}$

ところで、 $((A \cup e B) \cap e C) \in PR(E)$ であり要素間には互いに素であるから、 $Y_t = Y_{t+1}$ となる。

(10) E_1, E_n は P_i の任意の要素であるので、結局、 P_i の全要素が Y_t に含まれ P_i の全要素の集合和である W_i は、 Y_t に含まれる。つまり、 W_i に対して、 $W_i \subseteq Y_t$ なる $Y_t \in ((A \cup e B) \cap e C)$ が存在する。

(11) 結論

(7)と(10)より、(a) が証明された。(証明終り)

・ 変数使用式集合 (VEXP[s,v])

ある特定の中間命令 s において変数 v の値が更新されるとき影響を受ける式の集合を、中間命令 s における変数 v の変数使用式集合と呼び、VEXP[s,v]と表現する。形式的には次の数式 1 6 に示すような式の集合である。

{ 数式 1 6 }

$VEXP[s,v] = \{e \mid e \text{ は } E \text{ に属する次の形の式である。但し、} op \text{ はアドレス演算子 \& 以外の任意の演算子である。}$

【 0 0 5 7 】・ 変数 v

・ 式「 op v」 (v をオペランドとする単項式)

・ 式「 v op b」 または、式「 b op v」 (v をオペランドとする 2 項式)

・ 式「 *p」 但し、 $v \in POINT[s,p]$

}

(第 1.3 章 本実施形態にて特別に用いる集合)

・ E_GEN 集合

基本ブロック B における $E_GEN[B]$ は次に示すような等価式集合の集合である。また、別ないい方をすると、基本ブロック B において発生する等価な関係のうち、基本ブロック B の出口に到達する。 $E_GEN[B]$ を求めるアルゴリズムを図 1 4 に示す。

【 0 0 5 8 】但し図 1 4 において、 $WSET$ は式の集合を保持する作業用の変数であり、また、a は変数かまたは間接演算式、b は式(変数、定数、単項式、2 項式)である。図 1 4 の文 (1) は簡単にいうと、変数 a を使用する式を、 $WSET$ に属する集合 X の要素から取り去ることを意味する。また文 (2) は、ポインタ変数 p が指す変数 v を使用する式を、 $WSET$ に属する集合 X の要素から取り去ることを意味する。文 (3) での処理は式 a が属する集合 Y が既に

30

集合 $WSET$ の要素として存在するなら、Y に式 b を追加することを意味する。文 (4) での処理は式 b が属する集合 Y が既に集合 $WSET$ の要素として存在するなら、Y に式 a を追加することを意味する。文 (5) は新たに式 a, b を要素とする等価式集合を $WSET$ に追加することを意味する。文 (6) は、関数 f の呼び出しにより変更される可能性のある変数 v を使用する式を、 $WSET$ に属する集合 X の要素から取り去ることを意味する。

【 0 0 5 9 】次に図 1 2 の基本ブロック B1 における集合 $WSET$ の推移を示す。但し、図 1 2 の中間命令 s4 におけるポインタ変数 d が指す変数は変数 a, b とする。すなわち $POINT[s4,d] = \{a,b\}$ 。また中間命令 s6 の関数 f の呼び出しにより値が変化する変数は x とする。すなわち $CHANGE[s6,f] = \{x\}$ 。まず中間命令 s1 に対する処理では、図 1 4 (5) において $\{(a,b+10)\}$ が $WSET$ に加わり、 $WSET = \{(a,b+10)\}$ となる。

【 0 0 6 0 】次に中間命令 s2 に対する処理では、図 1 4 (1) において (a, b+10) から式 b+10 が削除されて、 $WSET = \{(a)\}$ となり、次に図 1 4 (5) において $\{(b,x+20)\}$ が $WSET$ に加わり、 $WSET = \{(a), (b,x+20)\}$ となる。次に中間命令 s3 に対する処理では、図 1 4 (5) において $\{(c,x+y)\}$ が $WSET$ に加わり、 $WSET = \{(a), (b,x+20), (c,x+y)\}$ となる。

【 0 0 6 1 】次に中間命令 s4 に対する処理では、図 1 4 (2) において a, b を使用している式を含む (a), (b, x+20) から a, b が削除されて、 $WSET = \{(x+20), (c,x+y)\}$ となり、次に図 1 4 (5) において $\{(*d,30)\}$ が $WSET$ に加わり、 $WSET = \{(x+20), (c,x+y), (*d,30)\}$ となる。次に中間命令 s5 に対する処理では、図 1 4 (4) において (e, x+y) が $WSET$ の要素 (c, x+y) に加わり、 $WSET = \{(x+20), (c,x+y), (*d,30)\}$ となる。

【 0 0 6 2 】次に中間命令 s6 に対する処理では、図 1 4 (6) において変数 x を使用している式が $WSET$ から削除され、 $WSET = \{(c,e), (*d,30)\}$ となる。次に中間命令 s7 に対する処理では、図 1 4 (4) において (z, *d) が $WSET$ の要素 (*d,30) に加わり、 $WSET = \{(c,e), (*d,30,z)\}$ となる。次に中間命令 s8 に対する処理では、図 1 4 (1) において変数 a の値が変更されるので、(*d,30,z) から *d が削除され、 $WSET = \{(c,e), (30,z)\}$ となり、図 1 4 (5) において $\{(a,x+z)\}$ が $WSET$ に加わり、 $WSET = \{(c,e), (30,z), (a,x+z)\}$ となる。

【 0 0 6 3 】以上から最終的な $E_GEN[B1]$ の値は、 $\{(c,e), (30,z), (a,x+z)\}$ となる。尚、 $E_GEN[B]$ の計算アルゴリズムから解るように、代入演算による等価な変数は同じ集合に属することになるので、それぞれの集合間には共通項が存在せず互いに素であるので、 $E_GEN[B]$ は集合 $PR(E)$ に属する。

・ E_PRE 集合

全式集合のうち、基本ブロック B 内の各中間命令における変数の更新によって影響を受けない式の集合である。 $E_PRE[B]$ を求めるアルゴリズムを図 1 5 に示す。

50

(第1.4章 本実施形態にて特別に用いるデータフロー方程式)

・等価式集合に関するデータフロー方程式

上述の用語を用いて、基本ブロック B の入口、出口に到達する式間の等価な関係を表す、等価式集合群 $E_IN[B]$ 、 $E_OUT[B]$ に関するデータフロー方程式を次の数式17に示す。

{ 数式17 }

$$E_IN[B] = \bigcap_{P \in \text{pred}[B]} E_OUT[P]$$

$$P \in \text{pred}[B]$$

$$E_OUT[B] = E_GEN[B] \cup (E_IN[B] \cap \{E_PRE[B]\})$$

この方程式を「等価な関係の到達」という観点から簡単に説明すると、 $E_IN[B]$ は各先行ブロックの出口における等価な関係のうち、全ての出口で成立しているもののみが、基本ブロック B の入口に到達することを意味する。 $E_OUT[B]$ は、基本ブロック B の出口に到達する等価な関係は、基本ブロック B で発生する等価な関係 $E_GEN[B]$ と、基本ブロック B の入口に到達する等価な関係のうち B で変更されないものであることを意味する。

・関数 $f(X) = E_GEN[B] \cup (X \cap \{E_PRE[B]\})$ の単調性の証明

前述の到達する定義の説明でも触れたように、前述の等価式集合に関するデータフロー方程式を、後述する通り *

$$f(X \cap Y) = G \cup ((X \cap Y) \cap P) \dots (1)$$

$$f(X) \cap f(Y) = (G \cup (X \cap P)) \cap (G \cup (Y \cap P)) \dots (2)$$

ここで、(2)の右辺をさらに変形する。

【0065】(2)の右辺の左側から順に「G」をA部、※「 $(X \cap P)$ 」をB部、「 $(G \cup (Y \cap P))$ 」をC部と考

$$(G \cap (G \cup (Y \cap P))) \cup ((X \cap P) \cap (G \cup (Y \cap P)))$$

... (3)

さらに、 \cap の可換性から整理すると次のようになる。 30

$$(3) = (G \cup (Y \cap P)) \cap G \cup ((G \cup (Y \cap P)) \cap (X \cap P)) \\ = S$$

(3)の式の左側から順に最初の「G」をA部、次の「 $(Y \cap P)$ 」をB部、次の「G」をC部、その次の「G」をA部、次の「 $(Y \cap P)$ 」をB部、次の「 $(X \cap P)$ 」をC部★

$$(G \cap G) \cup ((Y \cap P) \cap G) \cup (G \cap (X \cap P)) \cup ((Y \cap P) \cap (X \cap P))$$

... (4)

ここで、演算 \cap の同一性、交換性、連合性から整理すると次のようになる。

$$(4) = (G \cup (G \cap (Y \cap P))) \cup ((G \cap (X \cap P)) \cup ((X \cap P) \cap (Y \cap P)))$$

さらに、演算 \cup の交換性、連合性から整理すると次の

$$(4) = f(X \cap Y) \cup ((G \cap Y \cap P) \cup (G \cap X \cap P)) \\ = T$$

となる。ここで、一般に次のことが成り立つ。

【0067】

$$X, Y \in PR(E), \forall x_i \in X, \exists z_j \in (X \cup Y), x_i \subseteq z_j.$$

よって、

$$\forall a_i \in f(X \cap Y), \exists t_j \in T, a_i \subseteq t_j.$$

* 返しアルゴリズムで解くには、文献 [2] p79-p88に示すように、合流演算「 \cap 」において、集合 $PR(E)$ がセミラテスであることと（これについては既に示した）、前述の等価式集合に関するデータフロー方程式において、基本ブロック B における影響を表す $E_OUT[B]$ の右辺の $E_IN[B]$ を X とおいた関数、 $f(X) = E_GEN[B] \cup (X \cap \{E_PRE[B]\})$ に単調性が成り立つことを証明する必要がある。ここでは、簡単化のために $E_GEN[B]$ 、 $\{E_PRE[B]\}$ をそれぞれ G 、 P とし、 $f(X) = G \cup (X \cap P)$ の単調性の証明について述べる。

【0064】文献 [3] p81-83より、単調性の証明は次の数式18を証明する必要がある。

{ 数式18 }

$$X, Y \in PR(E)$$

$$f(X \cap Y) \cap f(Y) = (f(X) \cap f(Y)) = f(X \cap Y)$$

さらに、このことを示すには、文献 [3] p81より、次の数式19を証明できれば十分である。

{ 数式19 }

任意の $A_i \in f(X \cap Y)$ 、に対して、 $B_j \in f(X) \cap f(Y)$ が存在して、

$$A_i \subseteq B_j$$

[証明] まず、 $f(X \cap Y)$ および $f(X) \cap f(Y)$ に実際の関数 f を適用すると次のようになる。

※「 $(X \cap P)$ 」をB部、「 $(G \cup (Y \cap P))$ 」をC部と考えて、C部分をA,B部分に分配すると次のようになる。

★と考えてC部分をA,B部分に分配すると次のようになる。

【0066】

☆ようになる。

$$(4) = (G \cup ((X \cap Y) \cap P)) \cup ((G \cap (Y \cap P)) \cup (G \cap (X \cap P)))$$

ここで、(4)の $G \cup ((X \cap Y) \cap P)$ は、(1)より $f(X \cap Y)$ であるから、

である。さらに、前述の「・演算子 \cup への演算子 \cap の分配」の結果から、次のことが成り立つ。

【0068】 $\forall t_i \in T, \exists s_j \in S, t_i \subseteq s_j$

$$\forall s_i \in S, \exists b_j \in (f(X) \cap f(Y)), s_i \subseteq b_j$$

50 故に、

$\forall A_i \in f(X \cap e Y), \exists B_j \in (f(X) \cap e f(Y)), A_i \subseteq B_j$
が成立する。(証明終り)

(第1.5章 本実施形態にて特別に用いるアルゴリズム)

・等価式集合の繰り返しアルゴリズム

等価式集合に関するデータフロー方程式を解く、繰り返しアルゴリズムを図16に示す。

【0069】図16のアルゴリズムは、1つのブロックでも $E_OUT[B]$ の値が変化しなくなるまで、全ての基本ブロック B について計算を繰り返し、結果が終了した(つまり収束した)ときの $E_IN[B], E_OUT[B]$ の値が上記データフロー方程式の解となる。また、直観的には図16(1)は、基本ブロック B の出口で成り立つ等価関係として、 $E_GEN[B]$ 集合で示される基本ブロック B 内で発生する等価関係の他に、 $E_PRE[B]$ 集合のみを要素とする式集合群を等価和演算で追加することにより、基本ブロック B で更新されない式は全て等価関係にあると仮定している。これは、最終的により大きな等価集合を得るため、つまり式のデータフロー方程式を満たす最大解を得るためである。

・等価式置換アルゴリズム

等価式集合の繰り返しアルゴリズムにより求めた等価式集合を用いた、最適化の一例である等価式置換アルゴリズムを図17に示す。この処理は各基本ブロック単位に実行される。

(第2.1章 実施形態の総説)以下、本明細書に添付した図面を参照しながら、最適化装置の実施形態を説明する。ところで当業者で慣用されている最適化装置の流通・売買の形態は、最適化装置の機能を実現する実行形式のソフトウェアを記録媒体に記録して、パッケージソフトとして流通・売買する形態である。このようなパッケージソフトはこれを購買した顧客により汎用コンピュータにインストールされ、インストールされた汎用コンピュータがこのソフトウェア通りの処理を行うことにより汎用コンピュータは最適化装置としての機能を果たす。

【0070】上記のような形態が慣用されることを考えると、最適化装置としての機能主体は、汎用コンピュータが具備しているプロセッサ、メモリ等のハードウェア資源というよりは、記録媒体に記録されたソフトウェアと考える方が妥当である。また複雑な処理内容を有するソフトウェアは、複数のサブルーチンやワークエリアにて構成されることが一般的であるので、個々のサブルーチンやワークエリアが独立した構成要素と考えるべきである。以降の説明では、最適化装置の機能を果たすためにどのような機能を果たすサブルーチン、ワークエリアを構成要素として開発すべきかについて述べる。尚、最適化装置の構成要素の全てを新規に開発する必要は無く、既存のオペレーティングシステムやコンパイラ、最適化装置等がライブラリ化しているものを利用することも一般的である。このように既存のサブルーチン、ワー

クエリアを利用して実現される構成要素については詳細は述べないこととする。

(第2.2章 最適化装置の内部構成の全体像)図1は、本発明の実施例における最適化装置1の内部構成の全体像を示す図である。図1に示すように最適化装置は、最適化制御部11、制御フロー情報解析部12、データフロー情報解析部13、等価式情報解析部14、中間コード最適化部15、等価式置換最適化部16、制御フロー情報保持部17、データフロー情報保持部18、等価式情報保持部19、作業用式集合保持部10より構成される。これらの構成要素の逐次解説を行う。

【0071】最適化制御部11は、最適化全体を制御する。制御フロー情報解析部12は、最適化制御部11より起動され、従来と同様に、制御の流れが一方向である基本ブロックに中間プログラムを分割し、基本ブロック間の制御の流れを示す、制御フロー情報を求める。データフロー情報解析部13は、最適化制御部11より起動され、従来と同様に、前述の制御フロー情報を使用して、さらに中間コードを解析し、到達する定義、利用可能式などのデータフロー情報を求める。

【0072】中間コード最適化部15は、最適化制御部11より起動され、従来と同様に制御フロー情報とデータフロー情報を使用して、中間コードの最適化を行なう。データフロー情報保持部18は、前述のデータフロー情報解析部13で求めたデータフロー情報を保持する。制御フロー情報保持部17は、前述の制御フロー情報解析部12で求めた制御フロー情報を保持する。

【0073】等価式情報保持部19は、前述の等価式情報解析部14の結果である基本ブロックの入口、出口に達する等価集合の集合を保持する。等価式情報解析部14は、本最適化装置1の主要部分であり、最適化制御部11より起動され、前述の図16「等価式集合の繰り返しアルゴリズム」を用いて大域的な依存関係の解析を、最適化の前工程として行う。

(第2.2.1章 等価式情報解析部14を実現する全ての構成要素)本最適化装置における解析工程の特徴部を実現する等価式情報解析部14に焦点を絞った説明を行う。等価式情報解析部14の内部構成を図2に示す。等価式情報解析部14は、等価式情報生成制御部21、全体式集合検出部22、全式集合保持部23、等価式集合生成部24、 E_GEN 集合生成部25、変数使用式判定部26、 E_PRE 集合生成部27、 E_GEN 作業用式集合保持部28、 E_PRE 作業用式集合保持部29、 E_GEN 集合保持部30、 E_PRE 集合保持部31、 E_IN 集合保持部32、 E_OUT 集合保持部33、状態保持部34、旧 E_OUT 集合保持部35より構成される。等価式情報解析部14の構成要素の逐次解説を行う。

(第2.2.2章 等価式情報生成制御部21、全体式集合検出部22、全式集合保持部23の解説)等価式情報生成制御部21は、等価式情報解析部14の全体を制御す

る。

【0074】全式集合検出部22は、等価式情報生成制御部21より起動され、プログラムに現れる全ての式を検出し、全式集合保持部23に格納する。全式集合保持部23はプログラムに現れる全ての式を保持する。

(第2.2.3章 等価式集合生成部24の解説) 等価式集合生成部24は、等価式情報生成制御部21より起動され、 E_GEN 集合、 E_PRE 集合、 E_OUT 集合、 E_IN 集合を各基本ブロック毎について生成する。等価式集合生成部24による等価式集合生成は、大きく分けて3つのフェーズからなる。第1フェーズでは、各基本ブロックの E_GEN 集合及び E_PRE 集合のみを生成し、第2フェーズでは、第1フェーズにおいて生成された E_GEN 集合及び E_PRE 集合を用いて、 E_OUT 集合及び E_IN 集合の初期化を行う。第3フェーズでは、{数式17}のデータフロー方程式を繰り返し解法を用いて解くことにより、生成された E_OUT 集合及び E_IN 集合の最大解を導出する。

(第2.2.4章 等価式集合生成部24の処理内容の階層構造) 図27及び図28のフローチャートに上記等価式集合の処理内容を示す。

【0075】本フローチャートにおいて第1フェーズに相当するのは、ステップd2における E_GEN 集合生成部25による $E_GEN[B]$ の生成と、ステップd3における E_PRE 集合生成部27による $E_PRE[B]$ の生成とを全ての基本ブロックBについて繰り返すループ1である。フローチャートにおいて第2フェーズに相当するのは、初期化ブロックB1の E_OUT 集合及び E_IN 集合の初期化を行うステップd4と、等価積演算 $E_GEN[B2] \cap E_PRE[B2]$ を行うステップd6を初期ブロック以外の全ての基本ブロックB2について繰り返すループ2である。

【0076】第3フェーズに相当するのは、 $E_IN[B3] = \cap e E_OUT[B4]$ 、 $B4 \in \text{pred}[B3]$ の演算を行うステップd9と、 $E_OUT[B3]$ を旧 E_OUT 集合保持部35に格納するステップd10と、 $E_OUT[B3] = E_GEN[B3] \cup e (E_IN[B3] \cap e E_PRE[B3])$ の演算を行うステップd11と、 $E_OUT[B3]$ と、旧 E_OUT 集合保持部35との内容比較を行うステップd12と、内容比較の結果に応じて状態保持部34を「真」に設定するステップd13とを全ての基本ブロックについて繰り返し行うループ3と、状態保持部の設定が「真」に切り換わるまでループ3の起動を繰り返させるステップd14とである。本フローチャートにおける各ステップの逐次解説は後段で行う。

(第2.2.5章 E_GEN 集合生成部25の解説) E_GEN 集合生成部25は、図27のフローチャートのループ1が一回巡る度に、等価式集合生成部24より起動され、前述の図14の「 $E_GEN[B]$ の計算アルゴリズム」を用いて、基本ブロックの E_GEN 集合を生成する。図19から図23までのフローチャートに E_GEN 集合生成部25の処理内容を示す。 E_GEN 集合生成部25の処理内容は、最上位層、中間層、最下位層からなる階層構造を有する。

(第2.2.5.1章 E_GEN 集合生成部25の処理内容の階層構造) 図19のフローチャートは E_GEN 集合生成部25の階層構造における最上位層に位置し、基本ブロックに含まれている中間命令sを解析するためのループ構造になっている。ループ構造に相当するのは、中間命令sが式「 $a=b$ 」の様式であるかを判定するステップb3と、中間命令sが関数呼び出しであるかを判定するステップb19とを基本ブロックに属する全ての中間命令sについて繰り返すループ1であり、このループ1の繰り返しの過程で E_GEN 作業用式集合保持部28内に得られた集合を E_GEN 集合としている。

【0077】図20のフローチャートは、 E_GEN 集合生成部25の階層構造における中間層に位置し、最上位層における解析結果を受けて、更に深い内容の解析を行うと共に、 E_GEN 集合の生成を行うものである。その「深い内容の解析」に相当するのは、ステップb3において判定された式aが変数v1であるかを判定するステップb4と、式aが間接演算式「*p」であるかを判定するステップb9であり、「 E_GEN 集合の生成」に相当するのは、変数v1でも、間接演算式「*p」でも無い場合に、式集合Y(\forall 式集合 $y \in E_GEN$ 作業用式集合保持部28の保持集合)に式a及び式bが属しているかを判定するステップb16を E_GEN 作業用式集合保持部28に保持されている全ての式集合Yについて繰り返すループ7と、ループ7の繰り返しが継続されている期間において、一度でもステップb16での判定がYesと判定されると、式集合Yに式a及び式bを加えてループ7から抜け出て、最上位層である図19のフローチャートにリターンするステップb17と、ループ7における処理が全ての式集合Yについて繰り返されると、式a及び式bを要素とする式集合(a, b)を E_GEN 作業用式集合保持部28に格納するステップb18とである。

【0078】図21のフローチャートは、 E_GEN 集合生成部25の階層構造における最下位層に位置し、中間層における解析結果が「式aが変数v1」である場合のみ E_GEN 集合の生成を行うものである。「 E_GEN 集合の生成」に相当するのは、ステップb4において判別された変数v1の値を中間命令sにおいて更新した場合に、式e(\forall 式e \in 式集合x)が影響を受けるかを判定するステップb7と、影響を受ける場合に式集合xから式eを削除するステップb8とを式集合x(\forall 式集合x $\in E_GEN$ 作業用式集合保持部28の保持集合)内の全ての式xについて繰り返すループ3と、当該ループ3を E_GEN 作業用式集合保持部28に保持されている全ての式集合xについて繰り返すループ2とである。

【0079】図22のフローチャートは、 E_GEN 集合生成部25の階層構造における最下位層に位置し、中間層における解析結果が「式aが間接演算式「*p」」である場合のみ E_GEN 集合の生成を行うものである。変数v2(\forall 変数v2 $\in \text{POINT}[s, p]$)の値を中間命令sにおいて更

新した場合に、式 e (\forall 式 $e \in$ 式集合 x) が影響を受けるかを判定するステップb 1 3 と、影響を受けると判定されれば式集合 x から式 e を削除するステップb 1 4 とを備え、当該ステップb 1 3、b 1 4 を式集合 x 内の全ての式 e について繰り返すループ6と、当該ループ6をE_GEN作業用式集合保持部2 8 に保持されている全ての式集合 x (\forall 式集合 $x \in$ E_GEN作業用式集合保持部2 8 の保持集合) について繰り返すループ5と、当該ループ5をPOINT[s,p]に属する全ての変数 v_2 について繰り返すループ4とを有する。

【0080】図23のフローチャートは、E_GEN集合生成部25の階層構造における最下位層に位置し、最上位層における解析結果が「中間命令 s が関数呼出し f 」である場合にのみ、E_GEN集合の生成を行うものである。本フローチャートは、変数 v_3 (\forall 変数 $v_3 \in$ CHANGE[s,f]) の値を中間命令 s において更新した場合に、式 e (\forall 式 $e \in$ 式集合 x) が影響を受けるかを判定するステップb 2 3 と、影響を受けると判定されれば式集合 x から式 e を削除するステップb 2 4 とを備え、当該ステップb 2 3 を式集合 x 内の全ての式 e について繰り返すループ10と、当該ループ10をE_GEN作業用式集合保持部2 8 に保持されている全ての式集合 x (\forall 式集合 $x \in$ E_GEN作業用式集合保持部2 8 の保持集合) について繰り返すループ9と、当該ループ8をCHANGE[s,f]に属する全ての変数 v_3 について繰り返すループ8とを有する。

(第2.2.6章 変数使用式判定部26の処理内容) 変数使用式判定部26は、等価式集合生成部24、E_GEN集合生成部25及びE_PRE集合生成部27より外部から中間命令 s 、変数 v 、式 a が第1引数～第3引数として指定されて起動される。このようにして起動されると、変数使用式判定部26は、第1引数として指定された中間命令において、第2引数として指定された変数の保持値が更新された場合、第3引数として指定された式が影響を受けるか否かを判定する。図18にその処理内容のフローチャートを示す。本フローチャートでは、ステップa 1において式 a がアドレス演算子&の式「< a」であるかを判定し、そうでなければステップa 2～ステップa 5を起動する。そうであればこれらステップa 2～ステップa 5をスキップしてステップa 6において自身を起動した等価式集合生成部24、E_GEN集合生成部25及びE_PRE集合生成部27の何れかに『No』を戻り値として返す。

【0081】ステップa 2～ステップa 5では、式 e の様式が変数 v そのものみであるか(1)、式 e の様式が単項演算子 op_1 の式「 $op_1 v$ 」であるか(2)、2項演算子 op_2 の式「 $acp_2 v$ 」または、「 $v op_2 a$ 」であるか(3)、間接参照演算子*の式「*p」であり、かつ指示変数集合POINT[s,p]に変数 v が属しているか(4)を判定する。式 e が上述した(1)から(4)までの何れのケースにも合致しなければステップa 6において自身の起動元に『No』を返して

起動元にリターンするが、一つのケースに合致すればステップa 7において自身の起動元に『Yes』を返して起動元にリターンする。

(第2.2.7章 E_PRE集合生成部27の処理内容の階層構造) E_PRE集合生成部27は、図27のフローチャートのループ1が一巡する度に、等価式集合生成部24より起動され、前述の図15の「E_PRE[B]の計算アルゴリズム」を用いて、基本ブロックのE_PRE集合を生成する。図24から図26にそのフローチャートを示す。E_PRE集合生成部27の処理内容は、最上位層、中間層、最下位層からなる階層構造である。

【0082】図24のフローチャートは、E_PRE集合生成部27の処理の最上位層に相当し、基本ブロックに含まれる各中間命令 s についての解析を行う。この解析処理に相当するのは、中間命令 s (\forall 中間命令 $s \in$ 基本ブロックB) が式「 $a=b$ 」の様式であるかを判定するステップc 3と、中間命令 s が関数呼び出しであるかを判定するステップc 13とを基本ブロックに属する全ての中間命令 s について繰り返すループ1であり、このループ1の繰り返しの過程でE_PRE作業用式集合保持部29内に得られた集合をE_PRE集合とするステップc 18を有している。

【0083】図25のフローチャートは、E_PRE集合生成部27の処理の最下位層に相当し、より深い内容の解析処理と、その解析結果に基づいた式集合の生成を行う。この『解析処理』に相当するのは、式 a が変数 v_1 であるかを判定するステップc 4と、式 a が間接演算式「*p」であるかを判定するステップb 9とである。ステップc 4において式 a が変数 v_1 と判定された場合に式集合の生成を行うのは、判別された変数 v_1 の値を中間命令 s において更新した場合に、式 e (\forall 式 $e \in$ E_PRE作業用式集合保持部29の保持集合) が影響を受けるかを判定するステップc 6と、影響を受けると判定した場合に式 e をE_PRE作業用式集合保持部29から削除するステップc 7とを有し、当該ステップc 6をE_PRE作業用式集合保持部29内の全ての式 e について繰り返すループ2である。

【0084】ステップc 11において式 a が間接演算式「*p」と判定された場合に式集合の生成を行うのは、変数 v_2 (\forall 変数 $v_2 \in$ POINT[s,p]) の値を中間命令 s において更新した場合に、式 e (\forall 式 $e \in$ E_PRE作業用式集合保持部29の保持集合) が影響を受けるかを判定するステップc 11と、影響を受けると判定されればE_PRE作業用式集合保持部29の保持集合から式 e を削除するステップc 12とを備え、当該ステップc 11をE_PRE作業用式集合保持部29の保持集合内の全ての式 e について繰り返すループ4と、当該ループ4をPOINT[s,p]に属する全ての変数 v_2 について繰り返すループ3である。

【0085】図26のフローチャートも、E_PRE集合生成部27の処理の最下位層に相当し、式 a が関数呼出し

「f」と判定された場合に式集合の生成を行う。この式集合生成処理に相当するのは、変数 $v3$ (\forall 変数 $v3 \in \text{CHANGE}(s, f)$)の値を中間命令 s において更新した場合に、式 e (\forall 式 $e \in E_PRE$ 作業用式集合保持部2 9の保持集合)が影響を受けるかを判定するステップc 1 6と、影響を受けると判定されれば E_PRE 作業用式集合保持部2 9の保持集合から式 e を削除するステップc 1 7とを備え、当該ステップc 1 6を E_PRE 作業用式集合保持部2 9の保持集合内の全ての式 e について繰り返すループ6、当該ループ6を $\text{CHANGE}(s, f)$ に属する全ての変数 $v3$ について繰り返すループ5である。

(第2.2.8章 E_GEN 作業用式集合保持部2 8が設けられていることの意義。) E_GEN 作業用式集合保持部2 8は、一つの基本ブロックの中の各中間命令の実行過程において等価性が成立している全ての式を集合化して保持する。

【0086】図3の基本ブロックBLK3内に位置する中間命令 $s9$ から中間命令 $s24$ までの一連の中間命令に対して E_GEN 集合生成部2 5が E_GEN 集合を生成しようとした際、 E_GEN 作業用式集合保持部2 8がどのような要素を集合化するかを図6 (a)に示す。図6 (a)では、 E_GEN 作業用式集合保持部2 8が中間命令 $s9$ に対して処理を行った場合、 E_GEN 集合の要素として、式『 $b6$ 』と式『 $a5+10$ 』とが追加されていることがわかる。 E_GEN 作業用式集合保持部2 8が中間命令 $s10$ に対して処理を行った場合、 E_GEN 集合の要素として、式『 $z6$ 』と式『 $p6$ 』とが追加されていることがわかる。

(第2.2.9章 E_PRE 作業用式集合保持部2 9が設けられていることの意義。) E_PRE 作業用式集合保持部2 9は、初期状態で全式集合を保持する。保持されている全式集合は、 E_PRE 集合生成部2 7による解析が進行するにつれ削除されてゆく。最終状態、即ち、基本ブロックの最後の中間命令に対して影響を受ける式を特定すると、 E_PRE 作業用式集合保持部2 9が保持している全ての式は E_PRE 集合の要素となる。図3の基本ブロックBLK3内に位置する中間命令 $s9$ から中間命令 $s24$ までの一連の中間命令に対して、 E_PRE 集合生成部2 7が E_PRE 集合を生成しようとした際、 E_PRE 集合にどのように生成されてゆくかを図6 (b)に示す。図6 (b)における中間命令 $s20$ の欄と、中間命令 $s21$ の欄とを比較した場合、中間命令 $s20$ の欄において記載されていた変数 $a4$ が中間命令 $s21$ の欄では削除されていることがわかる。これは、図3の基本ブロックBLK3の中間命令 $s21$ において変数 $a4$ が定義されているからであり、本変数は、中間命令 $s21$ により影響を受ける変数として削除されているのである。

(第2.2.10章 E_GEN 集合保持部3 0が設けられていることの意義) E_GEN 集合保持部3 0は、基本ブロックの何れかのものの E_GEN 集合を E_GEN 集合生成部2 5が生成すると、生成された E_GEN 集合を基本ブロック別に格納する。図3に示す基本ブロックBLK1~BLK6に対して E_GE

N 集合生成部2 5が E_GEN 集合を生成すると、生成された各基本ブロック毎の E_GEN 集合は、図5 (b)に示すように E_GEN 集合保持部3 0により保持される。

(第2.2.11章 E_PRE 集合保持部3 1が設けられていることの意義) E_PRE 集合保持部3 1は、基本ブロックのうち、何れかのものの E_PRE 集合を E_PRE 集合生成部2 7が生成すると、生成された E_PRE 集合を基本ブロック別に格納する。図3に示す基本ブロックBLK1~BLK6に対して E_PRE 集合生成部2 7が E_PRE 集合を生成すると、生成された各基本ブロック毎の E_PRE 集合は、図5 (c)に示すように E_PRE 集合保持部3 1により保持される。

(第2.2.12章 E_OUT 集合保持部3 3が設けられていることの意義) データフロー方程式の最大解を求めるために E_OUT 集合保持部3 3は、 E_GEN 集合保持部3 0が保持している E_GEN 集合と、 E_PRE 集合保持部3 1が保持している E_PRE 集合とから初期化された E_OUT 集合を各基本ブロック毎に保持する。図3に示す基本ブロックBLK1~BLK6についての E_OUT 集合の初期内容の一例を図7 (b)に示す。

【0087】 E_GEN 集合と、 E_PRE 集合とに加えて、 E_IN 集合保持部3 2が保持している E_IN 集合をデータフロー方程式を解く処理に適用することにより、データフロー方程式を満たす E_OUT 集合を各基本ブロック毎に保持する。一回目のループ処理の後、図3に示す基本ブロックBLK1~BLK6について生成された E_OUT 集合がどのように変化したかを図8 (c)に示す。収束状態の E_OUT 集合を図9 (b)に示す。

【0088】状態保持部3 4は、等価式集合生成部2 4の処理過程において、繰り返し処理の判定に使用する。 IE_OUT 集合保持部3 5は、等価式集合生成部2 4の処理過程において、状態保持部3 4に値を設定する際に使用する。

(第2.2.13章 E_IN 集合保持部3 2が設けられていることの意義) E_OUT 集合保持部3 3が新たに保持した E_OUT 集合をデータフロー方程式を解く処理に適用することにより、データフロー方程式を満たす E_OUT 集合を各基本ブロック毎に保持する。

【0089】一回目のループ処理の後、図3に示す基本ブロックBLK1~BLK6について生成された E_IN 集合がどのように変化したかを図8 (a)に示す。等価式集合生成部2 4による処理が収束し、最大解となる E_IN 集合を図9 (a)に示す。

(第2.3章 等価式置換最適化部1 6の解説) 等価式置換最適化部1 6は、本発明の主要部分であり、最適化制御部1 1部より起動され、前述の図1 7の「等価式置換アルゴリズム」を用いて、中間プログラムの最適化を行なう。図2 9から図3 4までのフローチャートは等価式置換最適化部1 6の処理内容を示し、階層構造を形成している。これらのフローチャートは、実際の最適化処理を含むと共に、その最適化処理に付随した等価式集合の改

訂作業を多く含んでいる。このような改訂作業を行うのは、最適化処理を行うと式集合の要素が変動するからである。

【0090】以上で等価式情報解析部14を実現する構成要素についての説明を終える。

(第2.3.1章 等価式置換最適化部16の処理内容の階層構造)図29のフローチャートは等価式置換最適化部16の階層構造における最上位層に位置し、基本ブロックに含まれている中間命令sを逐次解析するための二重ループ構造になっている。二重ループは、中間命令sが式「a=b」の様式であるかを判定するステップe4と、中間命令sが関数呼び出しであるかを判定するステップe26と、中間命令sが「if a cmp b goto label」の様式であるかを判定するステップe32とを基本ブロックに属する全ての中間命令sについて繰り返すループ2と、このループ3を全ての基本ブロックについて繰り返すループ1とにより形成される。ここでcmpは条件演算子一般を意味している。

【0091】図30のフローチャートは、等価式置換最適化部16の階層構造における中間層に位置し、最上位層において中間命令sの様式が式「a=b」の様式と解析された場合に最適化を行い、その解析作業に付随した式集合の改訂作業を行う階層である。『最適化作業』に相当するのは、作業用式集合保持部101に式「a=b」の両辺が保持されているかを判定するステップe5と、保持されている場合、中間命令sの両辺は等価性が成立しているとしてその中間命令s自体を削除するステップe6と、中間命令sの右边bの様式が2項演算式「b1 op b2」であるかを判定するステップe7と、その計算結果が定数に置き換え可能な場合にその置き換えを敢行するステップe8と、中間命令sの右边bの様式が単項演算式「op b1」であるかを判定するステップe9と、その計算結果が定数に置き換え可能な場合にその置き換えを敢行するステップe10とである。

【0092】『解析作業に付随した改訂作業』に相当するのは、式aが変数であれば図31の最下位層のフローチャートを呼び出すステップe11と、式aが間接演算式「*p」であれば図32の最下位層のフローチャートを呼び出すステップe16と、式集合Y(∀式集合Y∈作業用式集合保持部101の保持集合)に式a及び式bが属しているかを判定するステップe23を作業用式集合保持部101に保持されている全ての式集合Yについて繰り返すループ8と、ループ8の繰り返しが続けられている期間において、一度でもステップe23での判定がYesと判定されると、式集合Yに式a及び式bを加えてループ8から抜け出して、最上位層である図29のフローチャートにリターンするステップe24とである。

【0093】図31のフローチャートは、等価式置換最適化部16の階層構造における最下位層に位置し、中間層における解析結果が「式aが変数v1」である場合のみ

等価式集合の改訂作業を行うものである。「等価式集合の改訂作業」に相当するのは、変数v1の値を中間命令sにおいて更新した場合に、式e(∀式e∈式集合x)が影響を受けるかを判定するステップe14と、影響を受ける場合に式集合xから式eを削除するステップe15とを式集合x(∀式集合x∈作業用式集合保持部101の保持集合)内の全ての式eについて繰り返すループ4と、当該ループ4を作業用式集合保持部101に保持されている全ての式集合xについて繰り返すループ3とである。

【0094】図32のフローチャートは、等価式置換最適化部16の階層構造における最下位層に位置し、中間層における解析結果が式「式aが間接演算式「*p」」である場合のみ等価式集合の改訂作業を行うものである。変数v2(∀変数v2∈POINT[s,p])の値を中間命令sにおいて更新した場合に、式e(∀式e∈式集合x)が影響を受けるかを判定するステップe20と、影響を受けると判定されれば式集合xから式eを削除するステップe21とを備え、当該ステップe20、ステップe21を式集合x内の全ての式eについて繰り返すループ7と、当該ループ7を作業用式集合保持部101に保持されている全ての式集合x(∀式集合x∈作業用式集合保持部101の保持集合)について繰り返すループ6と、当該ループ6をPOINT[s,p]に属する全ての変数v2について繰り返すループ5とを有する。

【0095】図33のフローチャートは、等価式置換最適化部16の階層構造における最下位層に位置し、最上位層における解析結果が「中間命令sが関数呼出しf」である場合にのみ、等価式集合の改訂作業を行うものである。本フローチャートは、変数v3(∀変数v3∈CHANGE[s,f])の値を中間命令sにおいて更新した場合に、式e(∀式e∈式集合x)が影響を受けるかを判定するステップe30と、影響を受けると判定されれば式集合xから式eを削除するステップe31とを備え、当該ステップe30、e31を式集合x内の全ての式eについて繰り返すループ10と、当該ループ10を作業用式集合保持部101に保持されている全ての式集合x(∀式集合x∈作業用式集合保持部101の保持集合)について繰り返すループ9と、当該ループ8をCHANGE[s,f]に属する全ての変数v3について繰り返すループ8とを有する。

【0096】図34のフローチャートは、等価式置換最適化部16の階層構造における中間層に位置し、最上位層における解析結果が中間命令sが「if a cmp b goto label」である場合にのみ、条件式に対しての最適化を行うものである。本フローチャートは、条件式「a cmp b」の左辺式a及び右辺式bそれぞれと等価性が成立する式であって、上記条件式を恒真とする式a1,b1が式集合に存在するかを判定するステップe33と、そのような式のa1,b1が存在する場合、中間命令sの条件式「a cmp b」を「(1)」に置き換えるステップe34と、条件式「a cmp b」の左辺式a及び右辺式bのそれぞれと等価性

が成立する式であって、上記条件式を恒偽とする式 a_2, b_2 が式集合に存在するかを判定するステップe 3 5と、そのような式 a_2, b_2 が存在する場合、中間命令sの条件式「 $a \text{ cmp } b$ 」を「(0)」に置き換えるステップe 3 6とにより条件式「 $a \text{ cmp } b$ 」を最適化する。

(第2.4章 これまでに示したフローチャートの各ステップの逐次解説)変数使用式判定部2 6の処理過程を図1 8のフローチャートの逐次解説により更に具体化する。変数使用式判定部2 6の起動時の引数は、中間命令sと変数vと式eであり、中間命令sにおいて変数vの値を更新したときに、式eが影響を受けるとき「YES」を戻り値として起動元に返し、そうでないとき「NO」を返す。以下その詳細について説明する。但しaは任意の変数または定数とし、pはポインタ変数とする。

【0 0 9 7】ステップa 1では、式eがアドレス演算子&の式「 $\& a$ 」であるときステップa 6へ進む。ステップa 2では、式eの様式が変数vそのもののみであるときステップa 7へ進む。ステップa 3では、式eの様式が単項演算子op1の式「 $op1 \ v$ 」であるときステップa 7へ進む。

【0 0 9 8】ステップa 4では、式eの様式が2項演算子op2の式「 $a \ op2 \ v$ 」または、「 $v \ op2 \ a$ 」であるときステップa 7へ進む。ステップa 5では、式eの様式が間接参照演算子*の式「 $*p$ 」であり、かつ指示変数集合POINT[s,p]に変数vが属しているときはステップa 7へ進む。ステップa 6では、NOを返して変数使用式判定部を終了する。

【0 0 9 9】ステップa 7では、YESを返して変数使用式判定部を終了する。E_GEN集合生成部2 5の処理過程を図1 9から2 3のフローチャートの逐次解説により更に具体化する。E_GEN集合生成部2 5を起動する際の引数は基本ブロックBであり、E_GEN[B]を算出しこの結果を戻り値としてE_GEN集合保持部3 0に格納する。以下その詳細について説明する。

【0 1 0 0】ステップb 1では、E_GEN作業用式集合保持部2 8を空にする。ステップb 2では、基本ブロックの先頭から順に中間命令を取り出し、取り出した中間命令sについて、ステップb 3からステップb 2 4の処理を実行する。基本ブロックB内の全ての中間命令について、処理を終了したらステップb 2 5へ進む。

【0 1 0 1】ステップb 3では中間命令sが「 $a = b$ 」の様式であれば、ステップb 4からステップb 1 8を行ない、そうでないときはステップb 1 9へ進む。(但し、ここでa,bは式である。)

ステップb 4では、ステップb 3で判別された式aが変数v1のとき次のステップb 5へ進む。そうでないときステップb 9へ進む。

【0 1 0 2】ステップb 5では、E_GEN作業用式集合保持部2 8に格納されている式集合xを取り出し、取り出した式集合xについてステップb 6からステップb 8の

処理を実行する。E_GEN作業用式集合保持部2 8に格納されている全ての式集合について処理を終了したらステップb 1 5へ進む。ステップb 6では、ステップb 5で取り出した式集合xに属する式を取り出し、取り出した式eについて、次のステップb 7、8の処理を実行する。ステップb 5で取り出した式集合xに属する全ての式について処理を終了したら、ステップb 5へ進み次の式集合を取り出す。

【0 1 0 3】ステップb 7では、ステップb 2で取り出した中間命令sと、ステップb 4で判別された変数v1と、ステップb 6で取り出した式eとを入力として変数使用式判定部2 6を起動する。判定結果が「YES」であるとき次のステップb 8へ進む。そうでないときはステップb 6へ進み次の式を取り出す。ステップb 8では、ステップb 6で取り出した式eをステップb 5で取り出した式集合xから削除する。ステップb 6へ進み次の式を取り出す。

【0 1 0 4】ステップb 9では、ステップb 3で判別された式aが間接演算式「 $*p$ 」であるとき次のステップb 1 0へ進む。そうでないときは、ステップb 1 5へ進む。ステップb 1 0では、ステップb 9で判別されたポインタ変数pが指す変数集合POINT[s, p]に属する変数v2を取り出し、取り出した変数v2について、ステップb 1 1からステップb 1 4の処理を実行する。ステップb 9で判別された変数pが指す変数集合POINT[s,p]に属する全ての変数について処理を終了したらステップb 1 5へ進む。

【0 1 0 5】ステップb 1 1では、E_GEN作業用式集合保持部2 8に格納されている式集合を順次取り出し、取り出した式集合xについて、ステップb 1 2からステップb 1 4の処理を実行する。E_GEN作業用式集合保持部2 8に格納されている全ての式集合について処理を終了したら、ステップb 1 0へ進み次の変数を取り出す。ステップb 1 2では、ステップb 1 1で取り出した式集合xに属する式を順次取り出し、取り出した式eについてステップb 1 3、1 4の処理を実行する。ステップb 1 1で取り出した式集合xに属する全ての式について、処理を終了したらステップb 1 1へ進み次の式集合を取り出す。

【0 1 0 6】ステップb 1 3では、ステップb 2で取り出した中間命令sと、ステップb 1 0で判別された変数v2と、ステップb 1 2で取り出した式eとを入力として変数c変数使用式判定部2 6を起動する。判定結果が「YES」であるとき次のステップb 1 4へ進む。そうでないときはステップb 1 2へ進み次の式を取り出す。ステップb 1 4では、ステップb 1 2で取り出した式eをステップb 1 1で取り出した式集合xから削除する。ステップb 1 2へ進み次の式を取り出す。

【0 1 0 7】ステップb 1 5では、E_GEN作業用式集合保持部2 8に格納されている式集合を順次取り出し、取

45

り出した式集合 γ についてステップb 16、17の処理を実行する。 E_GEN 作業用式集合保持部28に格納されている全ての式集合について処理が終了したら、つまり、ステップb 16の判定に該当する式集合 γ が存在しなかった時、ステップb 18へ進む。ステップb 16では、ステップb 15で取り出した式集合 γ の中に、ステップb 3で判別された式 a または式 b が属しているとき、ステップb 17へ進む。そうでないときはステップb 15へ進み次の式集合を取り出す。

【0108】ステップb 17では、ステップb 15で取り出した式集合 γ にステップb 3で判別された式 a および式 b を加える。ステップb 2へ進み、次の中間命令を取り出す。ステップb 18では、ステップb 3で判別された式 a, b を要素とする式集合 (a, b) を生成し、 E_GEN 作業用式集合保持部28に格納する。ステップb 2へ進み、次の中間命令を取り出す。

【0109】ステップb 19では、ステップb 2で取り出した中間命令 s が関数 f の呼び出しであるとき、次のステップb 20に進み、そうでないときはステップb 2へ進み次の中間命令を取り出す。ステップb 20では、ステップb 19で判別された中間命令 s における関数 f の呼び出しにより変更される可能性のある 変更変数集合CHANGE $[s, f]$ に属する変数を順次取り出し、取り出した変数 v_3 について、ステップb 21からステップb 24の処理を実行する。変更変数集合CHANGE $[s, f]$ に属する全ての変数について、処理を終了したらステップb 2へ進み、次の中間命令について処理を行なう。

【0110】ステップb 21では、 E_GEN 作業用式集合保持部28に格納されている式集合 x を順次取り出し、取り出した式集合 x について、ステップb 22からステップb 24の処理を実行する。 E_GEN 作業用式集合保持部28に格納されている全ての式集合について、処理を終了したらステップb 20へ進み次の変数を取り出す。ステップb 22では、ステップb 21で取り出した式集合 x に属する式を順次取り出し、取り出した式 e について、次のステップb 23、24の処理を実行する。ステップb 21で取り出した式集合 x に属する全ての式について、処理を終了したらステップb 21へ進み次の式集合を取り出す。

【0111】ステップb 23では、ステップb 2で取り出した中間命令 s と、ステップb 20で判別された変数 v_3 と、ステップb 22で取り出した式 e とを入力として変数使用式判定部26を起動する。判定結果が「YES」であるとき次のステップb 24へ進む。そうでないときはステップb 22へ進み次の式を取り出す。ステップb 24では、ステップb 22で取り出した式 e をステップb 21で取り出した式集合 x から削除する。ステップb 22へ進み次の式を取り出す。

【0112】ステップb 25では、 E_GEN 作業用式集合保持部28の内容を基本ブロックBの $E_GEN[B]$ 集合とし

46

て E_GEN 集合保持部30に格納し、 E_GEN 集合生成部25の処理を終了する。 E_PRE 集合生成部27の処理過程を図24から26のフローチャートの逐次解説により更に具体化する。 E_PRE 集合生成部27を起動する際の引数は基本ブロックBであり、 $E_PRE[B]$ を算出しこの結果を戻り値として E_PRE 集合保持部31に格納する。以下その詳細について説明する。

【0113】ステップc 1では、全式集合保持部23に格納されている全ての式を、 E_PRE 作業用式集合保持部29に格納する。ステップc 2では、基本ブロックの先頭から順に中間命令を取り出し、取り出した中間命令 s について、ステップc 3からステップc 17の処理を実行する。基本ブロックB内の全ての中間命令について、処理を終了したらステップc 18へ進む。

【0114】ステップc 3では、中間命令 s が「 $a = b$ 」の形であれば、ステップc 4からステップc 12を行ない、そうでないときはステップc 13へ進む。(但し、ここで a, b は式である。)

ステップc 4では、ステップc 3で判別された式 a が変数 v_1 のとき次のステップc 5へ進む。そうでないときステップc 8へ進む。

【0115】ステップc 5では、 E_PRE 作業用式集合保持部29に属する式を取り出し、取り出した式 e について、次のステップc 6、7の処理を実行する。 E_PRE 作業用式集合保持部29に属する全ての式について処理を終了したら、ステップc 2へ進み次の中間命令を取り出す。ステップc 6では、ステップc 2で取り出した中間命令 s と、ステップc 4で判別された変数 v_1 と、ステップc 5で取り出した式 e とを入力として 変数使用式判定部26を起動する。判定結果が「YES」であるとき次のステップc 7へ進む。そうでないときはステップc 5へ進み次の式を取り出す。

【0116】ステップc 7では、ステップc 5で取り出した式 e を E_PRE 作業用式集合保持部29から削除する。ステップc 5へ進み次の式を取り出す。ステップc 8では、ステップc 3で判別された式 a が間接演算式「 $*p$ 」であるとき次のステップc 9へ進む。そうでないときは、ステップc 13へ進む。ステップc 9では、ステップc 8で判別された変数 p が指す変数集合POINT $[s, p]$ に属する変数 v_2 を取り出し、取り出した変数 v_2 について、ステップc 10からステップc 12の処理を実行する。ステップc 8で判別された変数 p が指す変数集合POINT $[s, p]$ に属する全ての変数について処理を終了したらステップc 2へ進み次の中間命令を取り出す。

【0117】ステップc 10では、 E_PRE 作業用式集合保持部29に属する式を順次取り出し、取り出した式 e についてステップc 11、12の処理を実行する。 E_PRE 作業用式集合保持部29に属する全ての式について、処理を終了したらステップc 9へ進み次の変数を取り出す。ステップc 11では、ステップc 2で取り出した中

間命令sと、ステップc 9 で判別された変数v2と、ステップc 10 で取り出した式eとを入力として変数使用式判定部26を起動する。判定結果が「YES」であるとき次のステップc 12へ進む。そうでないときはステップc 10へ進み次の式を取り出す。

【0118】ステップc 12では、ステップc 10で取り出した式eをE_{PRE}作業用式集合保持部29から削除する。ステップc 10へ進み次の式を取り出す。ステップc 13では、ステップc 2で取り出した中間命令sが関数fの呼び出しであるとき、次のステップc 14に進み、そうでないときはステップc 2へ進み次の中間命令を取り出す。

【0119】ステップc 14では、ステップc 13で判別された中間命令sにおける関数fの呼び出しにより変更される可能性のある変更変数集合CHANGE[s, f]に属する変数を順次取り出し、取り出した変数v3について、ステップc 15からステップc 17の処理を実行する。変更変数集合CHANGE[s, f]に属する全ての変数について、処理を終了したらステップc 2へ進み次の中間命令について処理を行なう。

【0120】ステップc 15では、E_{PRE}作業用式集合保持部29に属する式を順次取り出し、取り出した式eについてステップc 16、17の処理を実行する。E_{PRE}作業用式集合保持部29に属する全ての式について、処理を終了したらステップc 14へ進み次の変数を取り出す。ステップc 16では、ステップc 2で取り出した中間命令sと、ステップc 14で判別された変数v3と、ステップc 15で取り出した式eとを入力として変数使用式判定部26を起動する。判定結果が「YES」であるとき次のステップc 17へ進む。そうでないときはステップc 15へ進み次の式を取り出す。

【0121】ステップc 17では、ステップc 15で取り出した式eをE_{PRE}作業用式集合保持部29から削除する。ステップc 15へ進み次の式を取り出す。ステップc 18では、E_{PRE}作業用式集合保持部29の内容をE_{PRE}集合保持部31に格納し、E_{PRE}集合生成部27の処理を終了する。等価式集合生成部24の処理過程を図27、28のフローチャートの逐次解説により更に具体化する。

【0122】ステップd 1では、基本ブロックを順次取り出し、取り出した基本ブロックBについて、ステップd 2、3を行なう。全ての基本ブロックについて処理を終了したらステップd 4へ進む。ステップd 2では、基本ブロックBを入力としてE_{GEN}集合生成部25を起動し、生成した各基本ブロックBのE_{GEN}集合をE_{GEN}集合保持部30に格納する。

【0123】ステップd 3では、基本ブロックBを入力としてE_{PRE}集合生成部27を起動し、生成した各基本ブロックBのE_{PRE}集合をE_{PRE}集合保持部31に格納する。ステップd 4では、初期ブロックB1に関して、E_{IN}

集合保持部32に保持されるB1のE_{IN}集合を空集合にし、さらに、ステップd 2で算出されたE_{GEN}集合保持部30に保持されているB1のE_{GEN}集合を、E_{OUT}集合保持部33に保持されるB1のE_{OUT}集合に格納する。

【0124】ステップd 5では、初期ブロックB1以外の基本ブロックを順次取り出し、取り出した基本ブロックB2についてステップd 6の処理を実行する。初期基本ブロックB1以外の全ての基本ブロックについて処理を終了したらステップd 7へ進む。ステップd 6では、まず、全式集合保持部23に保持されている全ての式集合と、ステップd 3で算出されるE_{PRE}集合保持部31に保持されているB2のE_{PRE}集合に関して等価積演算を行なう。さらに演算結果と、ステップd 2で算出されるE_{GEN}集合保持部30に保持されているB2のE_{GEN}集合に関して、等価和演算を行ない、演算結果をE_{OUT}集合保持部33に保持される、基本ブロックB2のE_{OUT}集合に保持する。ステップd 5へ進み次の基本ブロックを取り出す。

【0125】ステップd 7では、状態保持部34に「偽」を格納する。ステップd 8では、基本ブロックを順次取り出し、取り出した基本ブロックB3について、ステップd 9からステップd 13を行なう。全ての基本ブロックについて処理を終了したらステップd 14へ進む。ステップd 9では、基本ブロックB3の全ての先行ブロックを取り出し、さらに取り出した先行ブロックB4のE_{OUT}集合保持部33に保持されている基本ブロックB4のE_{OUT}集合を取り出し、取り出したE_{OUT}集合(E_{OUT}[B4])について等価積演算を行ない、演算結果をE_{IN}集合保持部32に保持されるB3のE_{IN}集合に格納する。

【0126】ステップd 10では、E_{OUT}集合保持部33に保持されている基本ブロックB3のE_{OUT}集合を、旧E_{OUT}集合保持部35に格納する。ステップd 11では、まず、E_{IN}集合保持部32に保持されている基本ブロックB3のE_{IN}集合と、ステップd 3で算出されるE_{PRE}集合保持部31に保持されているB3のE_{PRE}集合に関して等価積演算を行なう。さらに得られた結果と、ステップd 2で算出されるE_{GEN}集合保持部30に保持されているB3のE_{GEN}集合に関して等価和演算を行ない、演算結果をE_{OUT}集合保持部33に保持されるB3のE_{OUT}集合に保持する。

【0127】ステップd 12では、旧E_{OUT}集合保持部35に格納されている内容と、E_{OUT}集合保持部33に保持されている基本ブロックB3のE_{OUT}集合の内容を比較し、異なっているときはステップd 13へ進み、同じ場合はステップd 8へ進み、次の基本ブロックを取り出す。ステップd 13では、状態保持部34に「真」を格納する。ステップd 8へ進み、次の基本ブロックを取り出す。

【0128】ステップd 14では、状態保持部34が「偽」であるか判定し、そうであるときステップd

10

20

30

40

50

ブ d 7 へ進む。そうでないとき、ステップ d 1 5 へ進む。ステップ d 1 5 では、E_IN 集合保持部 3 2、E_OUT 集合保持部 3 3 の内容を図 1 の等価式情報保持部 1 9 に格納し、等価式集合生成部の処理を終了する。等価式置換最適化部 1 6 の処理過程を図 2 9 から図 3 4 のフローチャートの逐次解説により更に具体化する。

【0129】ステップ e 1 では、基本ブロックを順次取り出し、取り出した基本ブロック B について、ステップ e 2 からステップ e 3 6 を行なう。全ての基本ブロックについて処理を終了したら等価式置換最適化部 1 6 の処理を終了する。ステップ e 2 では、等価式情報保持部 1 9 に格納されている基本ブロック B の E_IN 集合を取り出し、作業用式集合保持部 1 0 1 に格納する。

【0130】ステップ e 3 では、基本ブロックの先頭から順に中間命令を取り出し、取り出した中間命令 s について、ステップ e 4 からステップ e 3 6 の処理を実行する。基本ブロック B 内の全ての中間命令について、処理を終了したらステップ e 1 へ進み次の基本ブロックを取り出す。ステップ e 4 では、ステップ e 3 で取り出した中間命令 s が「a = b」の形であれば、ステップ e 5 からステップ e 2 4 を行ない、そうでないときはステップ e 2 5 へ進む。(但し、ここで a, b は式である。)

ステップ e 5 では、ステップ e 4 で判別された式 a, b の両方を要素とする式集合が、作業用式集合保持部 1 0 1 に格納されているとき、ステップ e 6 へ進む。そうでないときはステップ e 7 へ進む。

【0131】ステップ e 6 では、ステップ e 3 で取り出した中間命令 s をプログラム上から削除し、ステップ e 1 1 へ進む。ステップ e 7 では、ステップ e 4 で判別された式 b が 2 項演算「b1 op b2」であるとき、ステップ e 8 を行ない、そうでないときステップ e 9 に進む。ステップ e 8 では、ステップ e 7 で判定された式「b1 op b2」の、b1 が定数 c1 であるかまたは、作業用式集合保持部 1 0 1 に格納されている集合のうち b1 が属している集合に定数 c1 が含まれていて、かつ、b1 同様に、b2 が定数 c2 であるかまたは、作業用式集合保持部 1 0 1 に格納されている集合のうち b2 が属している集合に定数 c2 が含まれているとき「定数 c1 op 定数 c2」の計算結果を改めて式 b とする。またプログラム上の中間命令 s の式 b も「定数 c1 op 定数 c2」の計算結果に置き換える。次にステップ e 1 1 に進む。

【0132】ステップ e 9 では、ステップ e 4 で判別された式 b が単項演算「op b1」であるとき、ステップ e 1 0 を行ない、そうでないときステップ e 1 1 に進む。ステップ e 1 0 では、ステップ e 9 で判定された式「op b1」の、b1 が定数 c1 であるかまたは、作業用式集合保持部 1 0 1 に格納されている集合のうち b1 が属している集合に定数 c1 が含まれているとき「op 定数 c1」の計算結果を改めて式 b とする。またプログラム上の中間命令 s の式 b も「op 定数 c1」の計算結果に置き換える。

【0133】ステップ e 1 1 では、ステップ e 4 で判別された式 a が変数 v1 であるときステップ e 1 2 を行ない、そうでないときステップ e 1 6 を行なう。ステップ e 1 2 では、作業用式集合保持部 1 0 1 に格納されている式集合 x を取り出し、取り出した式集合 x についてステップ e 1 3 からステップ e 1 5 の処理を実行する。作業用式集合保持部 1 0 1 に格納されている全ての式集合について処理を終了したらステップ e 2 2 へ進む。

【0134】ステップ e 1 3 では、ステップ e 1 2 で取り出した式集合 x に属する式を取り出し、取り出した式 e について、次のステップ e 1 4、1 5 の処理を実行する。ステップ e 1 2 で取り出した式集合 x に属する全ての式について処理を終了したら、ステップ e 1 2 へ進み次の式集合を取り出す。ステップ e 1 4 では、ステップ e 3 で取り出した中間命令 s とステップ e 1 1 で判別された変数 v1 とステップ e 1 3 で取り出した式 e とを入力として変数使用式判定部 2 6 を起動する。判定結果が「YES」であるとき次のステップ e 1 5 へ進む。そうでないときはステップ e 1 3 へ進み次の式を取り出す。

【0135】ステップ e 1 5 では、ステップ e 1 3 で取り出した式 e をステップ e 1 2 で取り出した式集合 x から削除する。ステップ e 1 3 へ進み次の式を取り出す。ステップ e 1 6 では、式 a が間接演算式「*p」であるとき次のステップ e 1 7 へ進む。そうでないときは、ステップ e 2 2 へ進む。ステップ e 1 7 では、ステップ e 1 6 で判別された変数 p が指す変数集合 POINT[s, p] に属する変数 v2 を取り出し、取り出した変数 v2 について、ステップ e 1 8 からステップ e 2 1 の処理を実行する。ステップ e 1 6 で判別された変数 p が指す変数集合 POINT[s, p] に属する全ての変数について処理を終了したらステップ e 2 2 へ進む。

【0136】ステップ e 1 8 では、作業用式集合保持部 1 0 1 に格納されている式集合を順次取り出し、取り出した式集合 x について、ステップ e 1 9 からステップ e 2 1 の処理を実行する。作業用式集合保持部 1 0 1 に格納されている全ての式集合について処理を終了したら、ステップ e 1 7 へ進み次の変数を取り出す。ステップ e 1 9 では、ステップ e 1 8 で取り出した式集合 x に属する式を順次取り出し、取り出した式 e についてステップ e 2 0、2 1 の処理を実行する。ステップ e 1 8 で取り出した式集合 x に属する全ての式について、処理を終了したらステップ e 1 8 へ進み次の式集合を取り出す。

【0137】ステップ e 2 0 では、ステップ e 3 で取り出した中間命令 s とステップ e 1 7 で判別された変数 v2 と、ステップ e 1 9 で取り出した式 e とを入力として変数使用式判定部 2 6 を起動する。判定結果が「YES」であるとき次のステップ e 2 1 へ進む。そうでないときはステップ e 1 9 へ進み次の式を取り出す。ステップ e 2 1 では、ステップ e 1 9 で取り出した式 e をステップ e 1 8 で取り出した式集合 x から削除する。ステップ e 1

9へ進み次の式を取り出す。

【0138】ステップe22では、作業用式集合保持部101に格納されている式集合を順次取り出し、取り出した式集合xについてステップe23、24の処理を実行する。作業用式集合保持部101に格納されている全ての式集合について処理が終了したら、つまり、ステップe23の判定に該当する式集合xが存在しなかったとき、ステップe25へ進む。

【0139】ステップe23では、ステップe22で取り出した式集合xの中に、ステップe4で判別された式a10または式bが属しているとき、ステップe24へ進む。そうでないときはステップe22へ進み次の式集合を取り出す。ステップe24では、ステップe21で取り出した式集合xにステップe4で判別された式aおよび式bを加える。ステップe3へ進み次の中間命令を取り出す。

【0140】ステップe25では、ステップe4で判別された式a、bを要素とする式集合(a,b)を生成し、作業用式集合保持部101に格納する。ステップe3へ進み次の中間命令を取り出す。ステップe26では、ステップe3で取り出した中間命令sが関数fの呼び出しであるとき、ステップe27からステップe32を行ない、そうでないときはステップe33へ進む。

【0141】ステップe27では、ステップe26で判別された中間命令sにおける関数fの呼び出しにより変更される可能性のある変数集合CHANGE[s,f]に属する変数を順次取り出し、取り出した変数v3について、ステップe28からステップe32の処理を実行する。変更変数集合CHANGE[s,f]に属する全ての変数について、処理を終了したらステップe3へ進み、次の中間命令を取り出す。

【0142】ステップe28では、作業用式集合保持部101に格納されている式集合xを順次取り出し、取り出した式集合xについて、ステップe29からステップe32の処理を実行する。作業用式集合保持部101に格納されている全ての式集合について、処理を終了したらステップe27へ進み次の変数を取り出す。ステップe29では、ステップe28で取り出した式集合xに属する式を順次取り出し、取り出した式eについて、次のステップe30、31の処理を実行する。ステップe28で取り出した式集合xに属する全ての式について、処理を終了したらステップe28へ進み次の式集合を取り出す。

【0143】ステップe30では、ステップe3で取り出した中間命令sとステップe27で判別された変数v3と、ステップe29で取り出した式eとを入力として変数使用式判定部26を起動する。判定結果が「YES」であるとき次のステップe31へ進む。そうでないときはステップe29へ進み次の式を取り出す。ステップe31では、ステップe29で取り出した式eをステップe

28で取り出した式集合xから削除する。ステップe29へ進み次の式を取り出す。

【0144】ステップe32では、ステップe3で取り出した中間命令sが条件分岐「if a**cmp** b goto label1」(cmp: 条件演算子, label: 分岐先ラベル)であるとき、ステップe33からステップe34を行ない、そうでないときは、ステップe3へ進み次の中間命令を取り出す。ステップe33では、条件式「a1 **cmp** b1」を恒真とする、aと等価であるa1、bと等価であるb1が存在するとき、ステップe34を行ない、そうでないときステップe35へ進む。ここでa1は、aそのものまたは、作業用式集合保持部101に格納されているaが属する式集合に含まれているものであり、同様にb1もbそのものまたは、作業用式集合保持部101に格納されているbが属する式集合に含まれている。

【0145】ステップe34では、ステップe3で取り出した中間命令sのプログラム上の条件式を「(1)」と恒真を示す条件式に置き換える。ステップe3へ進み次の中間命令を取り出す。ステップe35では、条件式「a1 **cmp** b1」を恒偽とする、aと等価であるa1、bと等価であるb1が存在するとき、ステップe36を行ない、そうでないときステップe3へ進み次の中間命令を取り出す。ここでa1はaそのものまたは、作業用式集合保持部101に格納されているaが属する式集合に含まれているものであり、同様にb1もbそのものまたは、作業用式集合保持部101に格納されているbが属する式集合に含まれている。

【0146】ステップe36では、ステップe3で取り出した中間命令sのプログラム上の条件式を「(0)」と恒偽を示す条件式に置き換える。ステップe3へ進み次の中間命令を取り出す。

(第2.5章 最適化装置1の動作) 以上のように構成される本実施形態の最適化装置1における等価式情報解析部14と、等価式置換最適化部16について、具体的な処理内容をこれまでに示したフローチャートを参照しながら説明する。説明の前に以下では図3のプログラム例を用いる。図3において四角で囲んだものが基本ブロックを示している。基本ブロック内のs1: s2:などは中間命令を表す。その他の記述はC言語の記述に準ずる。また、図1の等価式情報解析部14が起動される前に、当該解析部で使用するデータが図1のデータフロー情報解析部13により算出される。具体的には図4に示すものであり、図4(a)は図3におけるポインタ変数p5のPOINT集合であり、図3の中間命令s22において変数a7を指し示すことを表している。また、図4(b)は図3における関数fの呼び出しに関するCHANGE集合であり、図3の中間命令s20において関数fにより更新される可能性のある変数がb6、z6であることを示している。これらの値は文献[1]p648-660で示されるような方法を用いて算出される。次に図1の等価式情報解析部14の処理内容を、

図2、図19から28のフローチャート、図3から41を参照しながら説明する。

【0147】まず、等価式情報解析部14において図2の等価式情報生成制御部21により全式集合検出部22が起動され、プログラム中に現れる条件式以外の全ての式が検出され全式集合保持部23に格納される。図3のプログラム例では図5(a)となる。次に等価式情報生成制御部21に等価式集合生成部24が起動され、最終的に各基本ブロック毎にE_IN集合、E_OUT集合が算出され図1の等価式情報保持部19に格納される。

【0148】[等価式集合生成部24の処理過程]次に等価式集合生成部24について図27、28フローチャートを用いて説明する。まず、ステップd1からd3において各基本ブロックのE_GEN集合、E_PRE集合が算出される。ここで、E_GEN集合生成部25とE_PRE集合生成部27の処理過程について説明する。

[E_GEN集合生成部25の処理過程]まず最初にE_GEN集合生成部25の処理過程を、図3の基本ブロックB3を例に、図18から23のフローチャートを用いて説明する。

【0149】まず図19のステップb1では、E_GEN作業用式集合保持部28を空にしその内容は図6(a)の最上段になる。次に図19のステップb2で中間命令s9が取り出され、ステップb3で中間命令s9が「a = b」の形であるので図20のステップb4に進み、ステップb4では式aが変数b6であるので図21のステップb5に進み、ステップb5ではE_GEN作業用式集合保持部28が空であるので図20のステップb15に進み、ステップb15ではE_GEN作業用式集合保持部28が空であるのでステップb18へ進み、ステップb18では図19のステップb3で判定された式a、bに相当する中間命令s9の式b6、a5+10から、式集合(b6,a5+10)が生成され、E_GEN作業用式集合保持部28に格納される。その結果、E_GEN作業用式集合保持部28の内容は図6(a)の中間命令s9の段になる。

【0150】次に図19のステップb2に戻り中間命令s10が取り出され、ステップb3で中間命令s10が「a = b」の形であるので図20のステップb4に進み、ステップb4では式aが変数z6であるので図21のステップb5に進み、ステップb5ではE_GEN作業用式集合保持部28から式集合(b6,a5+10)が取り出され、ステップb6では式集合(b6,a5+10)から式b6が取り出され、ステップb7では中間命令s10、変数z6、式b6を入力として変数使用式判定部26が起動される。次に変数使用式判定部26の図18のステップa1からa5では式b6はどの条件にも合致しないので、ステップa6で「NO」が起動先の図21のステップb7に返される。ステップb7では「NO」が返されたのでステップb6へ進み、次の式「a5+10」が取り出され、ステップb7では中間命令s10、変数z6、式「a5+10」を入力として変数使用式判定部

26が起動される。図18のステップa1からa5が実行されるが式「a5+10」はどの条件にも合致しないので、ステップa6が実行され「NO」が返される。図21のステップb7で再びステップb6に進むが取り出すべき式がないので、ステップb5へ進む。しかしステップb5でも取り出すべき次の式集合が、E_GEN作業用式集合保持部28にないので図20のステップb15へ進む。ステップb15ではE_GEN作業用式集合保持部28から式集合(b6,a5+10)が取り出され、ステップb16では式集合(b6,a5+10)に式z6、p6ともに属していないのでステップb15に進む。しかしステップb15では次の式集合が存在しないのでステップb18に進む。ステップb18では式集合(z6,p6)が生成され、E_GEN作業用式集合保持部28に格納される。その結果、E_GEN作業用式集合保持部28の内容は図6(a)の中間命令s10の段になる。

【0151】以下同様に中間ステップs11からs24まで処理した結果、E_GEN作業用式集合保持部28の内容は図6(a)の中間命令s24の段になる。特に図3の中間命令s18の処理では、変数b2が更新されるため図21のステップb8において、図6(a)の中間命令s17の段における式集合(a7,p7+b2)から式「p7+b2」が削除され、さらに、図20のステップb16において式「t21」が中間命令s17の段における式集合(t21,a)に属しているので、ステップb17で式集合(t21,a)に式「b2」が加えられる。また、図3の中間命令s20の処理では、中間命令s20が関数f1の呼び出しであるので、図19のステップb19の判定で「YES」となり、次に図23のステップb20からステップb24が実行されて、図4(b)の関数f1のCHANGE集合に属する変数b6、z6が関数f1の呼び出しによって更新される可能性があるため、図23のステップb24において、図6の中間命令19の段における式集合(z6,p6)、(b6,a5+10)から式z6,p6がそれぞれ削除される。また、図3中間命令s22の処理では、図20のステップb9の判定で「YES」と判定なり、次に図22のステップb10からステップb14が実行されて、図4(a)の変数p5の中間命令s22における指示変数集合に属する変数a7が更新される可能性があるため、図6の中間命令21の段における式集合(a7)、(z7,a7)から式a7が削除される。

【0152】次に図3の中間ステップs24は基本ブロックB3の最後の中間命令であるので、この中間命令に対する処理が終了と図19のステップb25が実行され、図6(a)の中間命令s24の段の内容がE_GEN集合保持部30に格納されて、図5(b)の基本ブロックBLK3の段の内容となる。以下同様に、基本ブロックBLK1,BLK2,BLK4,BLK5,BLK6に対してE_GEN集合生成部25が実行され、結果は図5(b)となる。

[E_PRE集合生成部27の処理過程]次に、E_PRE集合生成部27の処理過程を、図3の基本ブロックBLK3を例に、

図2 4 から2 6 のフローチャートを用いて説明する。

【 0 1 5 3 】まず最初に図2 4 のステップc 1 が実行されて、図6 (b)の最上段のように、図2 の全体式集合検出部2 2 で算出した 図5 (a)の全式集合保持部2 3 の内容を図2 のE_PRE作業用式集合保持部2 9 に格納する。次に図2 4 のステップc 2 で中間命令s9が取り出され、ステップc 3 で中間命令s5が「 a = b 」の形であるので図2 5 のステップc 4 に進み、ステップc 4 では式aが変数b6であるので同図のステップc 5 に進む、ステップc 5 ではE_PRE作業用式集合保持部2 9 に格納されている図6 (b)の最上段の式「 x5 」が取り出されステップc 6 に進み、ステップc 6 では中間命令s9、変数b6、式「 x5 」を入力として変数使用式判定部2 6 が起動され、図のステップa 1 からa 5 が実行されるが式「 x5 」はどの条件にも合致しないので、ステップa 6 が実行され「 NO 」が返される。同様に図6 (b)の最上段の残りの式に対してステップc 5 からステップ7 までの処理が行なわれる。特にE_PRE作業用式集合保持部2 9 に格納されている図6 (b)の最上段の式「 b6 」に対しては、ステップc 6 で変数使用式判定部2 6 の判定で「 YES 」が返され、ステップc 7 でE_PRE作業用式集合保持部2 9 から式「 b6 」が削除される。

【 0 1 5 4 】以下同様に中間ステップs10からs24まで処理した結果、E_PRE作業用式集合保持部2 9 の内容は図6 (b)の中間命令s24の段になる。特に、中間命令s18ではE_PRE作業用式集合保持部2 9 から図2 5 のステップc 7 において式「 b7+b2 」、「 b2 」が削除される。また中間命令s20では図4 (b)に示す関数f1のCHANGE集合に属する変数b6、z6が図2 6 のステップc 1 4 で取り出され、取り出された変数b6、z6それぞれに対して同様の処理が行なわれるので、E_PRE作業用式集合保持部2 9 から変数b6、z6を使用する式が削除される。但し、この例では既に中間命令s9、s10で変数b6、z6を使用する式が削除されているので、中間命令s20に関する処理によってE_PRE作業用式集合保持部2 9 の内容に変化はない。また中間命令s22では図4 (a)に示す変数p5の中間命令s22における指示変数集合に属する変数a7が図2 5 のステップc 9 で取り出され、取り出された変数a7に対して同様の処理が行なわれるので、E_PRE作業用式集合保持部2 9 から変数a7を使用する式が削除される。

【 0 1 5 5 】但し、この例では既に中間命令11で変数a7を使用する式が削除されているので、中間命令s22に関する処理によってE_PRE作業用式集合保持部2 9 の内容に変化はない。次に図3 の中間ステップs24は基本ブロックB 3 の最後の中間命令であるので、この中間命令に対する処理が終ると図2 4 のステップc 1 8 が実行され、図6 (b) の中間命令s24の段の内容がE_PRE集合保持部3 1 に格納されて、図5 (c)の基本ブロックBLK3の段の内容となる。

【 0 1 5 6 】以下同様に、基本ブロックBLK1、BLK2、BLK

4、BLK5、BLK6に対してE_PRE集合生成部2 7 が実行され、結果は図5 (c)となる。

[等価式集合生成部2 4 の処理過程の続き]次に等価式集合生成部2 4 の処理に戻って図2 7 ステップd 4 に進み、図3 の初期ブロックBLK1のE_IN集合が空集合にされ、さらにステップd 5、d 6 において基本ブロックBLK1以外の基本ブロックに関して、E_OUT集合の初期値を求められる。例えば基本ブロックBLK3では、図5 (b)の基本ブロックBLK3のE_GEN集合E_GEN[BLK3]と同図(c)のE_PRE集合E_PRE[BLK3]が取り出され、E_GEN[BLK3]と、E_PRE[BLK3]のみを要素とする集合{E_PRE[BLK3]}に対して等価演算が行なわれ、その結果図7 (b)の基本ブロックBLK3のE_OUT集合が得られる。同様にその他の基本ブロックに対して処理を行なった結果を図7 (b)に示す。

【 0 1 5 7 】次に図2 8 ステップd 7 に進み、図2 の状態保持部3 4 が「 偽 」にされ、ステップd 8 においてまず図3 の基本ブロックBLK1が取り出され、ステップd 9 では基本ブロックBLK1の先行ブロックは存在しないので、E_IN[BLK1]集合が空集合になり結果が図8 (a)のように基本ブロックBLK1の場所に格納される。ステップd 1 0 では現時点でのE_OUT[BLK1]集合が図3 の旧E_OUT集合保持部3 5 に格納される。よって図6 (c) (1)のように図7 (b)の基本ブロックBLK1のE_OUT集合が旧E_OUT集合保持部3 5 に格納される。次にステップd 1 1 では図7 (a)に示すとおり、基本ブロックBLK1のE_IN集合が空集合なので新しいE_OUT集合は図5 (b)の基本ブロックBLK1のE_GEN集合となる。よって図8 (c)のように基本ブロックBLK1の場所に格納される。次にステップd 1 2 では、図8 (c)の基本ブロックBLK1のE_OUT集合の内容と、図6 (c) (1)に示す旧E_OUT集合保持部3 5 の内容が比較され同じあるのでステップd 8 へ進む。

【 0 1 5 8 】次にステップd 8 では、次の基本ブロックBLK2が取り出され、ステップd 9 において基本ブロックBLK2の先行ブロックである基本ブロックBLK1とBLK5のE_OUT集合に対して等価演算が行なわれる。よって新たに求められて基本ブロックBLK1のE_OUT集合である図8 (c)の基本ブロックBLK1のE_OUT集合と、図7 (b)の基本ブロックBLK5のE_OUT集合の間で等価演算が行なわれ、結果が図8 (a)の基本ブロックBLK2のE_IN集合に格納される。

【 0 1 5 9 】次にステップd 1 0 では基本ブロックBLK2の旧E_OUT集合である図7 (b)の基本ブロックBLK2のE_OUT集合が、図6 (c) (2)のように旧E_OUT集合保持部3 5 に格納される。次にステップd 1 1 では、まず図8 (a)の基本ブロックBLK2のE_IN集合E_IN[BLK2]と図5 (c)の基本ブロックBLK2のE_PRE集合E_PRE[BLK2]が取り出され、集合E_IN[BLK2]と、集合E_PRE[BLK2]を唯一要素とする集合{E_PRE[BLK2]}に対して等価演算が行なわれる。その結果は図8 (b)の基本ブロックBLK2の段の内容とな

る。

【 0 1 6 0 】次に図5 (b)の基本ブロック BLK2のE_IN集合と、先に求めた図8 (b)の基本ブロック BLK2の段内容との間に等価演算が行なわれ、その結果が図8 (c)の基本ブロック BLK2のE_OUT集合に格納される。次にステップd 1 2では、図8 (c)の基本ブロック BLK2のE_OUT集合の内容と、図6 (c) (2)に示す旧E_OUT集合保持部3 5の内容が比較され、異なるのでステップd 1 3に進み状態保持部3 4が「真」に設定される。

【 0 1 6 1 】以下基本ブロック BLK3, BLK4, BLK5, BLK6 10 に対しても同様に処理が行なわれ、その結果は図8に示すとおりとなる。次にステップd 1 4に進み状態保持部3 4が「真」か判定し、「真」であるのでステップd 7に戻り、ステップd 8から再び全ての基本ブロックについてE_IN集合とE_OUT集合を再計算する。しかし、図3の例では、ステップd 8の2回目のループ処理において、基本ブロック BLK1のE_OUT集合に変化はなく、さらに、基本ブロック BLK1のE_OUT集合と基本ブロック BLK5のE_OUT集合の間で行なわれる等価演算の結果は、基本ブロック BLK2のE_IN集合と同じである。よって、基本 20 ブロック BLK2のE_OUT集合にも変化がなく、結局、基本ブロック BLK3, BLK4, BLK5, BLK6の各E_OUT集合にも変化がなくなるので、状態保持部3 4が「偽」のままとなる。よって、ステップd 8の2回目のループ処理が終了した後はステップd 1 2の判定が「NO」となり次にステップd 1 5に進み、ステップd 1 5では、図8 (a) (b)の内容が図9に示すように図1の等価式情報保持部1 9に格納され、等価式集合生成部2 4の処理が終了する。

【 0 1 6 2 】次に図1の等価式置換最適化部1 6の処理内容を、図3の基本ブロック BLK5, BLK4, BLK2に対する 30 処理について、図2 9からの図3 4のフローチャート、図9、4 2を参照しながら主要な要点を説明する。

[基本ブロック BLK2に対する処理の説明] まずはじめに、基本ブロック BLK2に対する処理について説明する。図2 9のステップe 2において図9 (a)の基本ブロック BLK2のE_IN集合 E_IN[BLK2]が図1の作業用式集合保持部 1 0 1へ格納され、図1 0 (a)の最上段の内容になる。次にステップe 3で図3の中間命令s5が取り出され、以後ステップe 4、図3 0のステップe 5、e 7、e 9、 40 e 1 1と実行され、さらに図3 1のステップe 1 2からe 1 5では中間命令s5の左辺である変数x4の更新によって影響を受ける作業用式集合保持部1 0 1に格納されている式集合が変更される可能性があるが、この場合は変数x4によっては変更されない。次に図3 0のステップe 2 2へ進み、ステップe 2 2からe 2 4が実行される。特に中間命令s5の変数p41が属する式集合 (p41, 10)が作業用式集合保持部1 0 1に格納されているので、ステップe 2 4において式集合 (p41, 10)に中間命令s5の変数x4が追加される。中間命令s5に関する処理の結果、作業用式集合保持部1 0 1の内容は図1 0 (a)の中間命令s5の 50

段となる。

【 0 1 6 3 】次に図3の中間命令s6が図2 9のステップe 3で取り出され、中間命令s5と同様に処理される。但し、中間命令s6の場合は図3 0のステップe 2 3において、作業用式集合保持部1 0 1に格納されている全ての式集合に関して「NO」と判定されるので、ステップe 2 5において式集合 (t32, x3)が作業用式集合保持部1 0 1に追加され、その結果、作業用式集合保持部1 0 1の内容は図1 0 (a)の中間命令s6の段となる。次に図3の中間命令s7が図2 9のステップe 3で取り出されるが、これも中間命令s5と同様に処理が施され、結果、作業用式集合保持部1 0 1の内容は図1 0 (a)の中間命令s7の段となる。

【 0 1 6 4 】次に図3の中間命令s8が図2 9のステップe 3で取り出され、中間命令s5と同様に処理が進むが、図2 9のステップe 3 2の判定で「YES」と判定され図3 4のステップe 3 3に進む。ここで、中間命令s8の式「b5」「x5」の両方が属する式集合 (x5, b5)が、図1 0 (a)の中間命令s7の段に存在し、変数x5とb5が等価であることが判明するので、条件式「b5 = x5」は恒真となる。よってステップe 3 4において図3のプログラム上の中間命令s8の条件式を恒真であることを示す「(1)」に置き換える。置き換えた結果を図1 1の中間命令s8に示す。

【 0 1 6 5 】次に基本ブロック BLK4, BLK5に関する処理は、原則的には基本ブロック BLK2と同等であるので、説明を省くためにその特徴的な処理部分についてのみ説明を行なう。

[基本ブロック BLK4に対する処理の説明] まず、図3の基本ブロック BLK4の中間命令s25からs27に関しては、基本ブロック BLK2の中間命令s5やs6と同様な処理が行なわれ、その結果、作業用式集合保持部1 0 1の内容は図1 0 (b)の中間命令s27の段となる。次に中間命令s28の処理では、ステップe 6において中間命令s28の式「x4+y4」が取り出され、変数x4, y4のそれぞれが、作業用式集合保持部1 0 1の中間命令s27の段の式集合 (x4, p41, 1 0), (y4, p42, 20)に属するので、変数x4, y4はそれぞれ定数10, 20と等価になり、結果、式「x4+y4」は定数30と等価となって図3の中間命令s28は図1 1の中間命令s28のように置き換えられる。

【 0 1 6 6 】[基本ブロック BLK5に対する処理の説明] 図3基本ブロック BLK5の中間命令s33を処理する前の作業用式集合保持部1 0 1の内容は図1 0 (c)の最上段となる。ここで図1 0 (c)の最上段には、中間命令s33の変数a1, b1の両方が属する式集合 (b1, a1)が存在するので、図3 0のステップe 5で「YES」と判定され、ステップe 6でプログラムから中間命令s33が冗長な中間命令として削除される。中間命令s34からs38の中間命令についても中間命令s33と同様に冗長な中間命令として削除されるので、その結果、図3の基本ブロック BLK5は図1

1の基本ブロックBLK5のように置き換えられる。

【0167】また、上記の説明で現れなかった図32、33のフローチャートに示す処理は、基本ブロックBLK3の中間命令s22およびs20のような間接参照式が代入の左辺に現れる中間命令や、関数呼び出しの中間命令のときに実行されるものであり、これは前述のE_GEN集合生成部25の処理過程で説明した図22、23の動作と同様な処理をする。

【0168】そして、残りの基本ブロックについても同様に処理されて、その結果、図3のプログラムは図11のように最適化される。このような本発明によれば、基本ブロックに跨って等価式集合を各中間命令毎に求めることが可能となり、等価式集合を使用してさらにプログラムの冗長性を削除するプログラム最適化が可能となる。

【0169】尚、図11の結果から中間命令s8の条件式が恒真であるので基本ブロックBLK3は実行されないことが解る。よって、図1の等価式置換最適化部16を実行した後に、再び、中間コード最適化部15を起動して従来の制御フローに対する最適化を行ない基本ブロックBLK3を削除してもよい。また、図1の等価式置換最適化部16を複数回繰り返してさらに最適化を行なってもよい。

【0170】また、図1の中間コード最適化部15に等価式置換最適化部16を組み入れて、中間コード最適化部15と等価式置換最適化部16を複数回数繰り返してさらに最適化を行なってもよい。

【0171】

【発明の効果】以上のように各命令の左辺及び右辺の何れか一方に位置する式を他の式に置き換えてもプログラムの実行が等価であることを示す式間の等価性を解析することにより、分岐元-分岐先の関係に基づいて複数の基本ブロックに分割されたプログラムを最適化する最適化装置であって、分岐元の基本ブロックの出口において複数の式間に成立している等価性であって、分岐先の基本ブロックの入口まで維持されるものを各基本ブロックについて解析する第1解析手段と、分岐先基本ブロックの入口にまで維持された等価性が、基本ブロック内に位置する先頭命令から終了命令までの実行によりどれだけ消滅するかを解析し、それと共に先頭命令から終了命令までの実行により基本ブロックの内部においてどれだけの等価性が新たに成立するかを各基本ブロックについて解析する第2解析手段と、第2解析手段による解析が行われると、第1解析手段による解析により基本ブロック入口において等価性成立が明らかになった式からなる集合である等価式集合E_INと、第2解析手段による解析により基本ブロック出口において等価性成立が明らかになった式からなる集合である等価式集合E_OUTとがその基本ブロックの最適化に用いられるだけの適格性を有するかを全ての基本ブロックについて判定する適格性判定手

段と、適格性を有しない場合、第1解析手段及び第2解析手段による解析工程を繰り返して行わせるよう第1解析手段及び第2解析手段を起動する繰返手段と、適格性を有する場合、第1解析手段及び第2解析手段の最後の起動により生成された等価式集合E_INを用いて基本ブロック内に位置する先頭命令から終了命令までを最適化するブロック内最適化手段によれば、複数基本ブロックの境界部の等価性発生-消滅をE_IN、E_OUT集合の形式で表しながら、式間の大域的な等価性を高精度に解析することができる。この繰返を何度も行えば、プログラム内の基本ブロックが如何なる実行順序で実行されても基本ブロックの入口で必ず等価性が成立する式を集合化することができ、これを利用した大域的な最適化を安全に行うことができる。

【0172】また、前記第1解析手段は、等価式集合E_OUTが生成されると、同一の基本ブロックを分岐先とする基本ブロック同士で等価式集合E_OUTの等価和演算を行うことにより、等価式集合E_INを再度生成する第1演算手段を備え、第2解析手段は、第1演算手段による演算により生成された等価式集合E_INと、等価式集合E_GEN生成手段により生成された等価式集合E_GENと、等価式集合E_PRE生成手段により生成された等価式集合E_PREとを以下の{数式1}に適用して等価式集合E_OUTを再度生成する第2演算手段とを備えることを特徴としている。

{数式1}

$E_OUT[B] = E_GEN[B] \cup_e (E_IN[B] \cap_e E_PRE[B])$

B: プログラム内の各基本ブロック

\cap_e : 等価積演算の演算子 \cup_e : 等価和演算の演算子

この構成によれば、等価性の変動が無くなるまでデータフロー方程式を解くために第1、第2演算手段は等価和演算、等価積演算を繰り返して行う。

【0173】データフロー方程式は、その単調性が保証された関数であるため、データフロー方程式を解く繰返し作業は有限時間内に完了することが保証されている。またデータフロー方程式は、全式集合のうち、各基本ブロックに含まれている先頭命令から最終命令までの実行により更新されないものからなる等価式集合E_PREと、各基本ブロックで生成される等価式集合E_GENとにより各基本ブロックのE_OUT集合が初期化されるので、データフロー方程式の繰返し作業により最大解を導き出すことができる。

【0174】各基本ブロックの等価式集合E_INに基づいて基本ブロック内に位置する各々の命令に対しての等価式集合が算出すれば、当該命令で使用される式と等価な式に置き換えることができ、冗長性が排除された命令への置き換えが可能となる。よってプログラムの冗長性が削減されて、結果的にコンパイラが生成する機械語プログラムの実行時間の短縮化や、プログラムサイズの縮小化が可能となる。

61

【0175】また、前記適格性判定手段は、第2 算出手段による演算により得られた等価式集合E_OUTが適格性を有するかを判定し、第1 演算手段は、等価式集合E_OUTが不適格であると判定されれば、同一の基本ブロックを分岐先とする基本ブロック同士の等価式集合E_OUTの等価和演算を行うことにより、等価式集合E_INを再度生成し、第2 演算手段は、第1 演算手段による演算により生成された等価式集合E_INと、等価式集合E_GEN生成手段により生成された等価式集合E_GENと、等価式集合E_PRE生成手段により生成された等価式集合E_PREとを前記
10 { 数式1 } に適用して等価式集合E_OUTを再度生成することを特徴としている。

【0176】また、前記適格性判定手段は、第2 算出手段による演算により得られた等価式集合E_OUTを過渡的な集合として記憶する過渡集合記憶部と、過渡集合を記憶した後、新たな等価式集合E_OUTが第2 算出手段による演算により得られると、両者を照合する照合部と、照合した結果、記憶されている過渡集合と、新たな等価式集合E_OUTとが一致すると、記憶された過渡集合が解としての適格性を満たすとの判定を下し、照合した結果、
20 記憶されている過渡集合と、新たな等価式集合E_OUTとが不一致であると、記憶された過渡集合が解として不適格であるとの判定を下す判定部とを備えることを特徴としている。この構成によれば、データフロー方程式を解く繰り返し作業の起動が適格性を満たすまで繰り返し行われるので、大域的な基本ブロックの最適化に求められる高精度な解を導き出せることができる。

【0177】前記等価式集合E_GEN生成手段は、等価性成立を解析すべき命令を取り出す第1 取出部と、各基本ブロックに含まれている先頭命令からそれまで取り出された命令までの間に等価性成立が明らかになった式の集合を保持しているE_GEN作業用保持部と、取り出された命令が代入であり、かつ代入の左辺が変数である場合、当該変数の代入により影響を受ける式を前記E_GEN作業用保持部が保持する式集合から削除する第1 のE_GEN作業用保持部更新部と、取り出された命令が代入であり、かつ代入の左辺が間接演算式である場合、当該間接演算式により更新される可能性のある変数によって影響を受ける式を前記E_GEN作業用保持部が保持する式集合から
40 削除する第2 のE_GEN作業用保持部更新部と、取り出された命令が代入であり前記E_GEN作業用保持部が保持する式集合の中に、当該命令における両辺のいずれかの式を含む式集合がある場合、当該命令の両辺の式を当該式集合に追加し、当該式集合が存在しないとき新たに当該命令の両辺を要素とする式集合を生成して前記E_GEN作業用保持部に追加する第3 のE_GEN作業用保持部更新部と、命令が関数呼び出し命令である場合、関数呼び出し命令によって更新される可能性のある変数によって影響を受ける式を前記E_GEN作業用保持部が保持する式集合から削除する第4 のE_GEN作業用保持部更新部とを備え
50

62

ることを特徴としている。

【0178】また、前記等価式集合E_GEN生成手段は、等価性成立を解析すべき命令を取り出す第1 取出部と、各基本ブロックに含まれている先頭命令からそれまで取り出された命令までの間に等価性成立が明らかになった式の集合を保持しているE_GEN作業用保持部と、取り出された命令が代入であり、かつ代入の左辺が変数である場合、当該変数の代入により影響を受ける式を前記E_GEN作業用保持部が保持する式集合から削除する第1 のE_GEN作業用保持部更新部と、取り出された命令が代入であり、かつ代入の左辺が間接演算式である場合、当該間接演算式により更新される可能性のある変数によって影響を受ける式を前記E_GEN作業用保持部が保持する式集合から削除する第2 のE_GEN作業用保持部更新部と、取り出された命令が代入であり前記E_GEN作業用保持部が保持する式集合の中に、当該命令における両辺のいずれかの式を含む式集合がある場合、当該命令の両辺の式を当該式集合に追加し、当該式集合が存在しないとき新たに当該命令の両辺を要素とする式集合を生成して前記E_GEN作業用保持部に追加する第3 のE_GEN作業用保持部更新部と、命令が関数呼び出し命令である場合、関数呼び出し命令によって更新される可能性のある変数によって影響を受ける式を前記E_GEN作業用保持部が保持する式集合から削除する第4 のE_GEN作業用保持部更新部とを備えることを特徴としている。この構成によれば、式間の等価関係を等価式集合のデータフロー解析で求めることが可能となり、等価関係の抽出を少ない時間で済ませることが可能となる。また、基本ブロックに跨った式間の等価関係が取得でき、より情報量の多い等価集合が得られるのでさらにプログラム全体の冗長性が削減されて、結果的にコンパイラが生成する機械語プログラムの実行時間の短縮化や、プログラムサイズの縮小化が可能となる。

【0179】また、ブロック内最適化手段は、初期状態において前記第1 算出手段により最後に算出された等価式集合E_INを保持し、同基本ブロック内の命令に対しての最適化が開始されると、その保持集合に対してこれまでに行われた同基本ブロック内の最適化に伴った更新がなされた等価式集合を保持している作業用保持部と、基本ブロックの先頭命令から末尾命令までを順番に取り出す第3 取出部と、取り出された式に対して最適化を行う命令最適化手段と、式が最適化されると、作業用保持部の保持集合を更新する更新手段とを備えることを特徴としている。この構成によれば、代入命令の両辺の式が等価関係にあるとき、当該代入命令が削除される。

【0180】また、前記ブロック内最適化手段は、等価式集合の集合を保持する作業用保持部を前記等価式集合E_INに設定する作業用保持部初期化手段を備え、命令最適化手段は、取り出された命令を、等価式集合保持部が保持している等価式集合内の式のうち、その式を等価性を有する式に置換することにより冗長性を削除する第1

の冗長性削除部と、取り出された命令が代入でありかつ代入の両辺の式が、前記作業用保持部が保持する同一の式集合に含まれている場合、プログラム上の当該命令を削除する第2の冗長性削除部とを備え、更新手段は、取り出された命令が代入でありかつ代入の左辺が変数である場合、当該変数の代入により影響を受ける式を前記作業用保持部が保持する式集合から削除する第1の作業用保持部更新部と、取り出された命令が代入でありかつ代入の左辺が間接演算式である場合、当該間接演算式により更新される可能性のある変数によって影響を受ける式を、前記作業用保持部が保持する式集合から削除する第2の作業用保持部更新部と、取り出された命令が代入でありかつ、前記作業用保持部が保持する式集合の中に、当該命令における両辺のいずれかの式を含む式集合がある場合、当該命令の両辺の式を当該式集合に追加し、当該式集合が存在しないとき新たに当該命令の両辺を要素とする式集合を生成し前記作業用保持部に追加する第3の作業用保持部更新部と、取り出された命令が関数呼び出し命令である場合、関数呼び出し命令によって更新される可能性のある変数によって影響を受ける式を、前記作業用保持部が保持する式集合から削除する第4の作業用保持部更新部と、前記作業用保持手段初期化手段を起動し、次に基本ブロック内の全ての命令に対して、前記第1、第2の冗長性削除部を起動しさらに、前記第1、2、3、4の作業用保持手段更新部を順番に起動する等価置換最適化制御部とを備えることを特徴としている。

【0181】この構成によれば、式の等価関係から可能であれば式を定数に置き換えて冗長が削除される。また、ブロック内最適化手段は、命令が代入である場合、右辺の式が前記作業用保持部が保持する式集合の中に含まれていて、当該式集合に定数が含まれているときは、当該定数で当該命令の右辺を置き換え、当該右辺の式が2項演算や単項演算であって当該演算で使用されている一つないし二つの変数が前記作業用保持部が保持する式集合の中に含まれており、かつ当該式集合に定数が含まれている場合、変数を定数に置き換えて当該演算を行ない、当該演算結果の定数値で当該命令の右辺を置き換える第3の冗長性削除部を有し、前記等価置換最適化制御部は、前記作業用保持部初期化手段を起動し、次に基本ブロック内の全ての命令に対して、前記第1、2、3の冗長性削除部を起動しさらに前記第1、2、3、4の作業用保持部更新部を順番に起動することを備えることを特徴としている。この構成によれば、条件式を可能であれば式の等価関係から恒真や恒偽に置き換えて冗長が削除され、また、請求項10から11の最適化装置によれば、基本ブロック内の最適化時に基本ブロック内の各命令の冗長性の削除とともに、各命令で更新される変数によって更新される式を考慮にいて、等価式集合を各命令毎に生成するので、より良い精度の高い等価式を各命令で求めることが可能であるので、結果的にコンパイラ

が生成する機械語プログラムの実行時間の短縮化や、プログラムサイズの縮小化が可能となる。

【図面の簡単な説明】

【図1】実施形態に係る最適化装置1の構成図である。

【図2】等価式情報解析部14の構成図である。

【図3】具体的な動作の説明に使用するプログラム例である。

【図4】(a) POINT集合の一例を示す図である。

(b) CHANGE集合の一例を示す図である。

【図5】(a) 全式集合保持部23の具体的な内容を示す図である。

(b) E_GEN集合保持部30の具体的な内容を示す図である。

(c) E_PRE集合保持部31の具体的な内容を示す図である。

【図6】(a) E_GEN作業用式集合保持部28の保持内容の推移を示す図である。

(b) E_PRE作業用式集合保持部29の保持内容の推移を示す図である。

(c) IE_OUT集合保持部35の保持内容の推移である。

【図7】(a) E_IN集合保持部32の初期内容を示す図である。

(b) E_OUT集合保持部33の初期内容を示す図である。

【図8】(a) E_IN集合保持部32の1回目のループ処理後の内容を示す図である。

(b) E_IN集合とE_PRE集合の間での等価積演算後の結果である。

(c) E_OUT集合保持部33の1回目のループ処理後の内容を示す図である。

【図9】(a) 等価式情報保持部19のE_IN集合の内容を示す図である。

(b) 等価式情報保持部19のE_OUT集合の内容を示す図である。

【図10】(a) 基本ブロックBLK2処理時の作業用式集合保持部101の保持内容の推移を示す図である。

(b) 基本ブロックBLK4処理時の作業用式集合保持部101の保持内容の推移を示す図である。

(c) 基本ブロックBLK5処理時の作業用式集合保持部101の保持内容の推移である。

【図11】図3のプログラムに対する等価式置換最適化部16を行なって最適化した後のプログラムである。

【図12】E_GEN集合の生成過程の例を示す説明図である。

【図13】(a) 指示変数集合の例を示す説明図である。

(b) 変更変数集合の例を示す説明図である。

【図14】E_GEN集合の計算アルゴリズムである。

【図15】E_PRE集合の計算アルゴリズムである。

【図16】等価式集合の計算アルゴリズムである。

【図17】等価式置換による最適化のアルゴリズムである。

【図18】変数使用式判定部26の処理のフローチャートである。

【図19】E_GEN集合生成部25の最上位層の処理内容を示すフローチャートである。

【図20】E_GEN集合生成部25の中間層の処理内容のフローチャートである。

【図21】E_GEN集合生成部25の最下位層の処理内容のフローチャートである。

【図22】E_GEN集合生成部25の最下位層の処理内容のフローチャートである。

【図23】E_GEN集合生成部25の最下位層の処理内容のフローチャートである。

【図24】E_PRE集合生成部27の最上位層の処理内容のフローチャートである。

【図25】E_PRE集合生成部27の中間層の処理内容のフローチャートである。

【図26】E_PRE集合生成部27の最下位層の処理内容のフローチャートである。

【図27】等価式集合生成部24の処理内容のフローチャートである。

【図28】等価式集合生成部24の処理内容のフローチャートである。

【図29】等価式置換最適化部16の最上位層の処理内容のフローチャートである。

【図30】等価式置換最適化部16の中間層の処理内容のフローチャートである。

【図31】等価式置換最適化部16の最下位層の処理内容のフローチャートである。

【図32】等価式置換最適化部16の最下位層の処理内容のフローチャートである。

【図33】等価式置換最適化部16の最下位層の処理内容のフローチャートである。

【図34】等価式置換最適化部16の中間層の処理内容のフローチャートである。

【図35】等価和演算の概念を示す概念図である。

【図36】従来のコンパイラの構成図である。

【図37】到達する定義の繰り返しアルゴリズムである。

【図38】従来技術の説明で用いるプログラム例である。

【図39】従来技術の説明で用いるプログラム例であ

る。

【図40】従来技術の説明で用いるプログラム例である。

【図41】従来技術の説明で用いるプログラム例である。

【図42】従来技術の説明で用いるプログラム例である。

【符号の説明】

1 最適化装置

10 11 最適化制御部

12 制御フロー情報解析部

13 データフロー情報解析部

14 等価式情報解析部

15 中間コード最適化部

16 等価式置換最適化部

17 制御フロー情報保持部

18 データフロー情報保持部

19 等価式情報保持部

101 作業用式集合保持部

20 21 等価式情報生成制御部

22 全体式集合検出部

23 全式集合保持部

24 等価式集合生成部

25 E_GEN集合生成部

26 変数使用式判定部

27 E_PRE集合生成部

28 E_GEN作業用式集合保持部

29 E_PRE作業用式集合保持部

30 E_GEN集合保持部

30 31 E_PRE集合保持部

32 E_IN集合保持部

33 E_OUT集合保持部

34 状態保持部

35 旧E_OUT集合保持部

41 構文解析装置

42 最適化装置

43 最適化制御部

44 制御フロー情報解析部

45 データフロー情報解析部

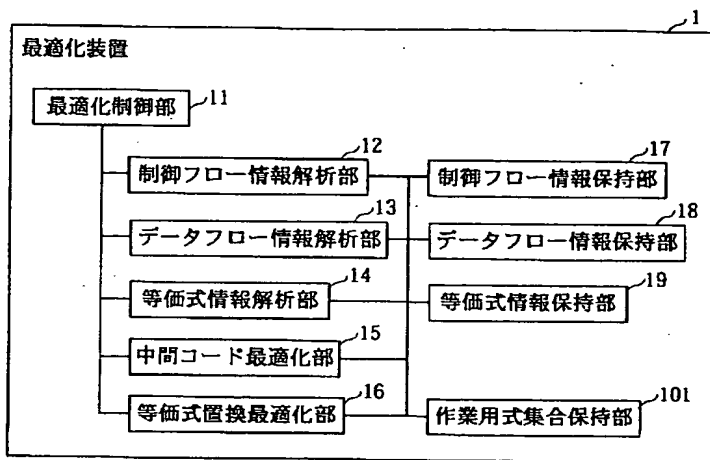
40 46 中間コード最適化部

47 制御フロー情報保持部

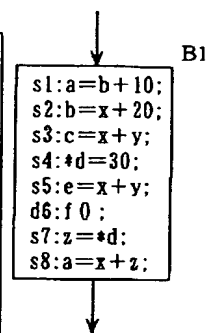
48 データフロー情報保持部

49 コード生成装置

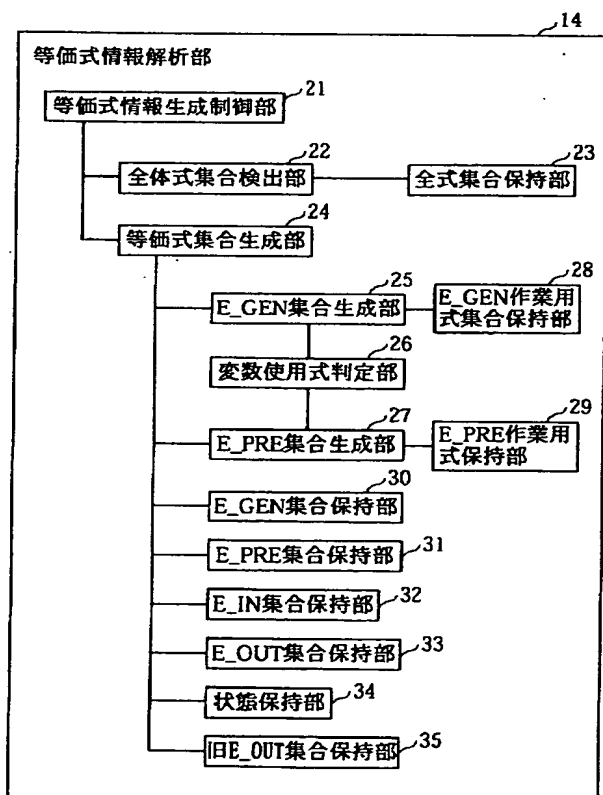
【 図1 】



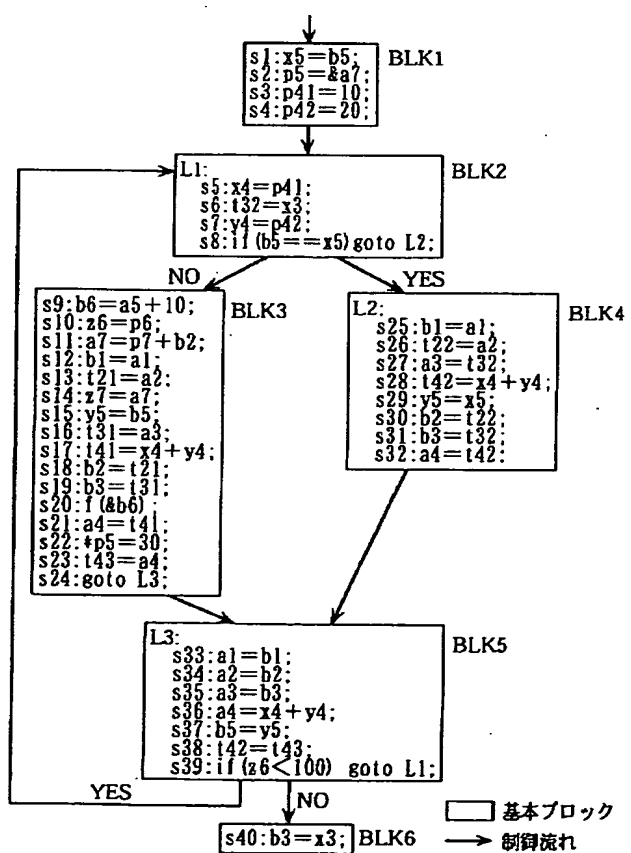
【 図12 】



【 図2 】



【 図3 】



【 図4 】

変数	中間命令	指示変数集合
p5	s5	(a7)
p5	s6	(a7)
p5	s7	(a7)
p5	s9	(a7)
p5	s10	(a7)
p5	s17	(a7)
p5	s18	(a7)
p5	s19	(a7)
p5	s20	(a7)
p5	s21	(a7)
p5	s22	(a7)
p5	s23	(a7)

(a) POINT集合

関数	中間命令	指示変数集合
f	s20	(b6, z6)

(b) CHANGE集合

【 図5 】

(x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, b6, a5+10, z6, p6, a7, p7+b2, b1, a1, t21, a2, z7, y5, b5, t31, a3, t41, x4+y4, b2, b3, b6, a4, p5, 30, t43, t22, t42)

(a) 全式集合保持部

基本ブロック	E_GEN集合
BLK1	{(x5, b5), (p5, a7), (p41, 10), (p42, 20)}
BLK2	{(x4, p41), (t32, x3), (y4, p42)}
BLK3	{(a5+10), (p6), (b1, a1), (t21, a2, b2), (x7), (y5, b5), (t31, a3, b3), (t41, x4+y4, a4, t43), (p5, 30)}
BLK4	{(b1, a1), (t22, a2, b2), (a3, t32, b3), (t42, x4+y4, a4), (y5, x5)}
BLK5	{(b1, b1), (a2, b2), (a3, b3), (a4, x4+y4), (b5, y5), (t42, t43)}
BLK6	{(b3, x3)}

(b) E_GEN集合保持部

基本ブロック	E_PRE集合
BLK1	{(b5, a7, 10, 20, x4, t32, x3, y4, b6, a5+10, z6, p6, a7, p7+b2, b1, a1, t21, a2, z7, y5, b5, t31, a3, t41, x4+y4, b2, b3, b6, a4, 30, t43, t22, t42)}
BLK2	{(x5, b5, p5, a7, p41, 10, p42, 20, x3, b6, a5+10, z6, p6, a7, p7+b2, b1, a1, t21, a2, z7, y5, b5, t31, a3, t41, b2, b3, b6, a4, p5, 30, t43, t22, t42)}
BLK3	{(x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, a5+10, p6, p7+b2, a1, a2, b5, a3, x4+y4, b6, p5, 30, t22, t42)}
BLK4	{(x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, b6, a5+10, z6, p6, a7, p7+b2, a1, t21, a2, z7, b5, t31, t41, x4+y4, b6, p5, 30, t43)}
BLK5	{(x5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, b5, a5+10, z6, p6, a7, p7+b2, b1, t21, z7, y5, b5, t31, t41, x4+y4, b2, b3, b6, p5, 30, t43, t22)}
BLK6	{(x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, b6, a5+10, z6, p6, a7, p7+b2, b1, a1, t21, a2, z7, y5, b5, t31, a3, t41, x4+y4, b2, b6, a4, p5, 30, t43, t22, t42)}

(c) E_PRE集合保持部

【 図9 】

基本ブロック	E_IN集合
BLK1	{(x5, b5), (p5, a7), (p41, 10), (p42, 20)}
BLK2	{(x4, p41, 10), (t32, x3), (y4, p42, 20), (x5, b5), (p5, a7)}
BLK3	{(x4, p41, 10), (t32, x3), (y4, p42, 20), (x5, b5), (p5, a7)}
BLK4	{(b1, a1), (a2, b2), (y5, b5), (a3, b3), (x4+y4, a4), (x4, p41, 10), (t32, x3), (y4, p42, 20), (x5, b5, y5), (p5, a7)}
BLK5	{(a1, b1), (a2, b2), (a3, b3), (a4, x4+y4), (b5, y5, x5), (t42, t43), (x4, p41, 10), (t32, x3), (y4, p42, 20), (p5, a7)}
BLK6	{(a1, b1), (a2, b2), (a3, b3), (a4, x4+y4), (b5, y5, x5), (t42, t43), (x4, p41, 10), (t32, x3), (y4, p42, 20), (p5, a7)}

(a) 等価式情報保持部のE_IN集合の内容

基本ブロック	E_OUT集合
BLK1	{(x5, b5), (p5, a7), (p41, 10), (p42, 20)}
BLK2	{(x4, p41, 10), (t32, x3), (y4, p42, 20), (x5, b5), (p5, a7)}
BLK3	{(a5+10), (p6), (b1, a1), (t21, a2, b2), (x7), (y5, b5, x5), (t31, a3, b3), (t41, x4+y4, a4, t43), (p5, 30), (x4, p41, 10), (t32, x3), (y4, p42, 20), (p5, a7)}
BLK4	{(b1, a1), (t22, a2, b2), (a3, t32, b3, x3), (t42, x4+y4, a4), (y5, x5, b5), (x4, p41, 10), (y4, p42, 20), (p5, a7)}
BLK5	{(a1, b1), (a2, b2), (a3, b3), (a4, x4+y4), (b5, y5, x5), (t42, t43), (x4, p41, 10), (t32, x3), (y4, p42, 20), (p5, a7)}
BLK6	{(a1, b1), (a2, b2), (a3), (a4, x4+y4), (b5, y5, x5), (t42, t43), (x4, p41, 10), (t32, x3, b3), (y4, p42, 20), (p5, a7)}

(b) 等価式情報保持部のE_OUT集合の内容

【 図15 】

[E_PRE[B]の計算アルゴリズム]

入力:基本ブロックB

出力:E_PRE[B]

方法:

WSET=E;

for (Bの全ての中間命令sについて繰り返す)

if (中間命令sが「a=b」であるとき)

if (aが変数v1のとき)

WSET=WSET-VEXP[s, v1];

... (1)

}

else

if (aが間接演算式*pであるとき)

for (全ての変数v2 ∈ POINT[s, p]について繰り返す)

(WSET=WSET-VEXP[s, v2];)

... (2)

}

else if (中間命令sが関数fの呼び出しであるとき)

for (全ての変数v3 ∈ CHANGE[s, f]について繰り返す)

(WSET=WSET-VEXP[s, v3];)

... (3)

}

return WSET;

【 図 6 】

中間命令	E_GEN作業用式集合保持部の内容
-	空
s9	$\{x5, p5+10, p6, p5\}$
s19	$\{x5, p5+10, p6, p5\}$
s17	$\{x5, p5+10, p6, p5\}, \{a7, p7+b2, b1, a1\}, \{t21, a2, b2, p7, a7\}, \{p5, b5\}, \{t31, a3, b3\}, \{t41, x4+y4, p4\}$
s18	$\{x5, p5+10, p6, p5\}, \{a7, p7+b2, b1, a1\}, \{t21, a2, b2, p7, a7\}, \{p5, b5\}, \{t31, a3, b3\}, \{t41, x4+y4, p4\}$
s20	$\{x5, p5+10, p6, p5\}, \{a7, p7+b2, b1, a1\}, \{t21, a2, b2, p7, a7\}, \{p5, b5\}, \{t31, a3, b3\}, \{t41, x4+y4, p4\}$
s21	$\{x5, p5+10, p6, p5\}, \{a7, p7+b2, b1, a1\}, \{t21, a2, b2, p7, a7\}, \{p5, b5\}, \{t31, a3, b3\}, \{t41, x4+y4, p4\}$
s22	$\{x5, p5+10, p6, p5\}, \{a7, p7+b2, b1, a1\}, \{t21, a2, b2, p7, a7\}, \{p5, b5\}, \{t31, a3, b3\}, \{t41, x4+y4, p4\}$
s23	$\{x5, p5+10, p6, p5\}, \{a7, p7+b2, b1, a1\}, \{t21, a2, b2, p7, a7\}, \{p5, b5\}, \{t31, a3, b3\}, \{t41, x4+y4, p4\}$
s24	$\{x5, p5+10, p6, p5\}, \{a7, p7+b2, b1, a1\}, \{t21, a2, b2, p7, a7\}, \{p5, b5\}, \{t31, a3, b3\}, \{t41, x4+y4, p4\}$

(a) E_GEN作業用式集合保持部の推移

中間命令	E_PRE作業用式集合保持部の内容
-	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, b6, a5+10, x6, p6, a7, p7+b2, b1, a1, t21, a2, z7, y5, b5, t31, a3, t41, x4+y4, b2, b3, b6, a4, p5, t43, t22, t42\}$
s9	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, b6, a5+10, x6, p6, a7, p7+b2, b1, a1, t21, a2, z7, y5, b5, t31, a3, t41, x4+y4, b2, b3, b6, a4, p5, t43, t22, t42\}$
s19	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, a5+10, p6, p7+b2, a1, a2, b5, a3, x4+y4, b6, a4, p5, t43, t22, t42\}$
s20	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, a5+10, p6, p7+b2, a1, a2, b5, a3, x4+y4, b6, a4, p5, t43, t22, t42\}$
s21	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, a5+10, p6, p7+b2, a1, a2, b5, a3, x4+y4, b6, a4, p5, t43, t22, t42\}$
s22	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, a5+10, p6, p7+b2, a1, a2, b5, a3, x4+y4, b6, a4, p5, t43, t22, t42\}$
s23	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, a5+10, p6, p7+b2, a1, a2, b5, a3, x4+y4, b6, a4, p5, t43, t22, t42\}$
s24	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, a5+10, p6, p7+b2, a1, a2, b5, a3, x4+y4, b6, a4, p5, t43, t22, t42\}$

(b) E_PRE作業用式集合保持部の推移

(1)	$\{x5, b5, p5, a7, p41, 10, p42, 20\}$
(2)	$\{x5, b5, p5, a7, p41, 10, p42, 20, x3, b5, a5+10, x6, p6, a7, p7+b2, b1, a1, t21, a2, z7, y5, b5, t31, a3, t41, b2, b3, b6, a4, p5, t43, t22, t42, x4, t32, y4\}$

(c) IBE_OUT集合保持部の推移

【 図 1 6 】

[等価式集合の計算アルゴリズム]

入力:制御フロー情報と各基本ブロックBについてのE_GEN[B]とE_PRE[B]
出力:E_OUT[B], E_IN[B]

方法:

for (全ての基本ブロックBについて繰り返す) {

 E_GEN[B]を計算する;

 E_PRE[B]を計算する;

}

E_IN[B1] = ϕ ; /*B1:初期ブロック*/

E_OUT[B1] = E_GEN[B1];

for (B1以外の全ての基本ブロックBについて繰り返す) {

 E_OUT[B] = E_GEN[B] \cup E_PRE[B]; ... (1)

}

repeat {

 changed=false;

 for (B1以外の全ての基本ブロックBについて繰り返す) {

 E_IN[B] = \cap E_OUT[B'];

 B' \in pred[B];

 old_E_OUT = E_OUT[B];

 E_OUT[B] = E_GEN[B] \cup E_IN[B] \cap E_PRE[B];

 if (old_E_OUT != E_OUT[B]) changed=true;

 }

until changed==false;

【 図 7 】

基本ブロック	E_IN集合
BLK1	\emptyset
BLK2	不定
BLK3	不定
BLK4	不定
BLK5	不定
BLK6	不定

(a) E_IN集合保持部の初期内容

基本ブロック	E_OUT集合
BLK1	$\{x5, b5\}, \{p5, a7\}, \{p41, 10\}, \{p42, 20\}$
BLK2	$\{x5, b5, p5, a7, p41, 10, p42, 20, x3, b6, a5+10, x6, p6, a7, p7+b2, b1, a1, t21, a2, z7, y5, b5, t31, a3, t41, b2, b3, b6, a4, p5, t43, t22, t42, x4, t32, y4\}$
BLK3	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, a5+10, p6, p7+b2, a1, a2, b5, a3, x4+y4, b6, a4, p5, t43, t22, t42, b1, t21, y5, t31, b3, t41, a4, t43\}, \{a7\}$
BLK4	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, b6, a5+10, x6, p6, a7, p7+b2, a1, t21, a2, z7, y5, t31, t41, x4+y4, b6, a4, p5, t43, b1, t22, b2, a3, t42, a4, y5\}$
BLK5	$\{x5, b5, a7, p41, 10, p42, 20, x4, t32, x3, y4, b6, a5+10, x6, p6, a7, p7+b2, b1, t21, z7, y5, b5, t31, t41, x4+y4, b2, b3, b6, a4, p5, t43, t22, a1, a2, a3, a4, t42\}$
BLK6	$\{x5, b5, p5, a7, p41, 10, p42, 20, x4, t32, x3, y4, b6, a5+10, x6, p6, a7, p7+b2, a1, t21, a2, z7, y5, b5, t31, a3, t41, x4+y4, b2, b6, a4, p5, t43, t22, t42, b3\}$

(b) E_OUT集合保持部の初期内容

【 図 1 4 】

[E_GEN[B]の計算アルゴリズム]

入力:基本ブロックB

出力:E_GEN[B]

方法:

WSET = ϕ ;

for (Bの先頭の中間命令から終りの中間命令まで、順に中間命令sを取り出し以下を繰り返す。) {

 if (中間命令sが「a=b」であるとき) {

 if (aが変数vのとき) {

 for (全ての式集合X \in WSETについて繰り返す) {

 X = X - VEIP[s, v]; ... (1)

 }

 } else

 if (aが間接演算式*であるとき) {

 for (全ての変数v2 \in POINT[s, p]について繰り返す) {

 for (全ての式集合X \in WSETについて繰り返す) {

 X = X - VEIP[s, v2];

 }

 ... (2)

 }

 if (Y \in WSET, a \in YなるYが存在するとき) Y = Y \cup {b}; ... (3)

 } else

 if (Y \in WSET, b \in YなるYが存在するとき) Y = Y \cup {a}; ... (4)

 } else WSET = WSET \cup {a, b}; ... (5)

 }

else if (中間命令sが関数fの呼び出しであるとき) {

 for (全ての式集合v3 \in CHANGE[s, f]について繰り返す) {

 for (全ての式集合X \in WSETについて繰り返す) {

 X = X - VEIP[s, v3];

 }

 } ... (6)

return WSET;

【 図8 】

基本ブロック	E_IN集合
BLK1	分岐先 E_OUT[BLK1] \cap e E_OUT[BLK5]
BLK2	(x5, b5), (p5, a7), (p41, 10), (p42, 20) ← 分岐元 分岐元
BLK3	(x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5), (p5, a7)
BLK4	(x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5), (p5, a7)
BLK5	(b1, a1), (a2, b2), (y5, b5), (a3, b3), (x4+y4, a4), (x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5, y5), (p5, a7)
BLK6	(a1, b1), (a2, b2), (a3, b3), (a4, x4+y4), (b5, y5, x5), (142, 143), (x4, p41, 10), (132, x3), (y4, p42, 20), (p5, a7)

(a) 1回目のループ処理後のE_IN集合保持部の内容

基本ブロック	E_IN[B] \cap e E_PRE[B]集合
BLK2	(x5, b5), (p5, a7), (p41, 10), (p42, 20) ← E_IN[BLK2] \cap e E_PRE[BLK2]
BLK3	(x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5), (p5, a7)
BLK4	(x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5), (p5, a7)
BLK5	(b1, a1), (a2, b2), (y5, b5), (a3, b3), (x4+y4, a4), (x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5, y5), (p5, a7)
BLK6	(a1, b1), (a2, b2), (a3, b3), (a4, x4+y4), (b5, y5, x5), (142, 143), (x4, p41, 10), (132, x3), (y4, p42, 20), (p5, a7)

(b) 1回目のループ処理時のE_IN[B] \cap e E_PRE[B]の算出結果

基本ブロック	E_OUT集合 E_GEN[BLK2] \cup e E_IN[BLK2] \cap e E_PRE[BLK2]
BLK1	(x5, b5), (p5, a7), (p41, 10), (p42, 20)
BLK2	(x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5), (p5, a7)
BLK3	(x5+y5, a5), (b1, a1), (121, a2, b2), (x7, y5, b5, x5), (131, a3, b3), (141, x4+y4, a4, 143), (p5, 30), (x4, p41, 10), (132, x3), (y4, p42, 20), (p5, a7)
BLK4	(b1, a1), (122, a2, b2), (a3, 132, b3, x3), (142, x4+y4, a4), (y5, x5, b5), (x4, p41, 10), (y4, p42, 20), (p5, a7)
BLK5	(a1, b1), (a2, b2), (a3, b3), (a4, x4+y4), (b5, y5, x5), (142, 143), (x4, p41, 10), (132, x3), (y4, p42, 20), (p5, a7)
BLK6	(a1, b1), (a2, b2), (a3, b3), (a4, x4+y4), (b5, y5, x5), (142, 143), (x4, p41, 10), (132, x3, b3), (y4, p42, 20), (p5, a7)

(c) 1回目のループ処理後のE_OUT集合保持部の初期内容

【 図10 】

中間命令	作業用式集合保持部の内容
—	(x5, b5), (p5, a7), (p41, 10), (p42, 20)
s5	(x5, b5), (p5, a7), (p41, 10, x4), (p42, 20)
s6	(x5, b5), (p5, a7), (p41, 10, x4), (p42, 20), (132, x3)
s7	(x5, b5), (p5, a7), (p41, 10, x4), (p42, 20, y4), (132, x3)
s8	(x5, b5), (p5, a7), (p41, 10, x4), (p42, 20, y4), (132, x3)

(a) 基本ブロックBLK2処理時の作業用式集合保持部の推移

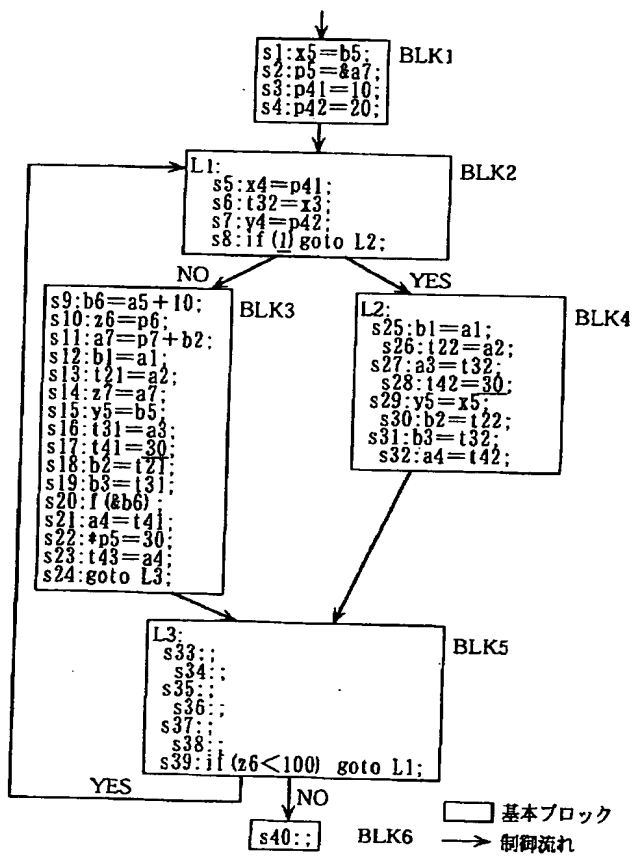
中間命令	作業用式集合保持部の内容
—	(x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5), (p5, a7)
s25	(x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5), (p5, a7), (b1, a1)
s26	(x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5), (p5, a7), (b1, a1), (122, a2)
s27	(x4, p41, 10), (132, x3, a3), (y4, p42, 20), (x5, b5), (p5, a7), (b1, a1), (122, a2)
s28	(x4, p41, 10), (132, x3, a3), (y4, p42, 20), (x5, b5), (p5, a7), (b1, a1), (122, a2), (142, 30)
---	-----

(b) 基本ブロックBLK4処理時の作業用式集合保持部の推移

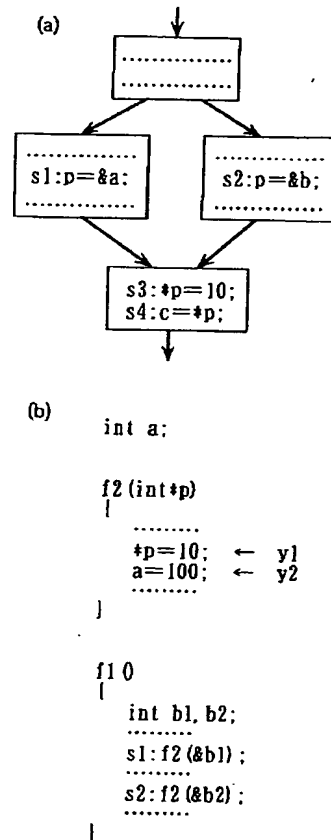
中間命令	作業用式集合保持部の内容
—	(b1, a1), (a2, b2), (y5, b5), (a3, b3), (x4+y4, a4), (x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5, y5), (p5, a7)
s33	(b1, a1), (a2, b2), (y5, b5), (a3, b3), (x4+y4, a4), (x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5, y5), (p5, a7)
s34	(b1, a1), (a2, b2), (y5, b5), (a3, b3), (x4+y4, a4), (x4, p41, 10), (132, x3), (y4, p42, 20), (x5, b5, y5), (p5, a7)
---	-----

(c) 基本ブロックBLK5処理時の作業用式集合保持部の推移

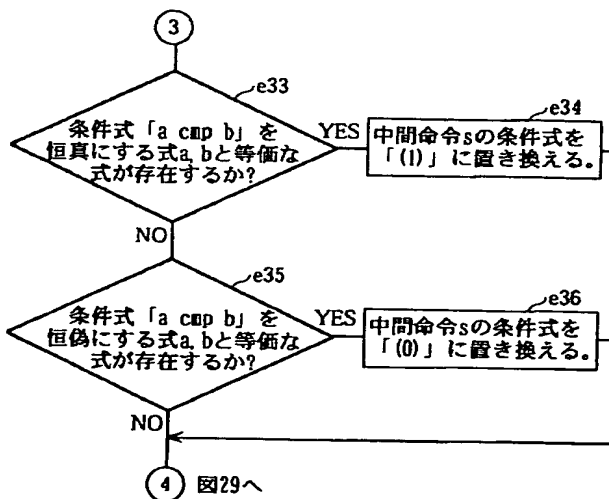
【 図11 】



【 図13 】



【 図34 】



【 図17 】

[等価式置換アルゴリズム]

入力:基本ブロックB

出力:等価式置換を行なった基本ブロックB

方法:

WSET←E_IN[B];

for (Bの先頭の間中命令から終りの中間命令まで、順に中間命令sを取り出し以下を繰り返す。)

if (中間命令sが「a=b」であるとき)

if (a∈X, b∈XなるX∈WSETが存在するとき) (中間命令sを削除する。)

else

if (bが2項演算であるとき、つまりbは「b1 op b2」であるとき)

if (「b1 op b2」が「定数c1 op 定数c2」に置き換え可能とき、但し、ここで、
c1はb1そのものまたは、X1∈WSET, b1∈X1において、c1∈X1なるものである。同様に、c2はb2そのものまたは、X2∈WSET, b2∈X2において、
c2∈X2なるものである。)

「定数c1 op 定数c2」の計算結果を改めて中間命令sの式bとする。

else

if (bが単項演算であるとき、つまりbは「op b1」であるとき)

if (「op b1」が「op 定数c1」に置き換え可能とき、但し、ここで、c1はb1
そのものまたは、X∈WSETにおいてb1∈Xかつ定数c1∈X1なる、定数c1
が存在するとき)

「op 定数c1」の計算結果を改めて中間命令sの式bとする。

if (aが変数v1のとき)

for (全てのX∈WSETについて繰り返す) {X=X-VEXP[s, v1];}

else

if (aが間接演算式*pであるとき)

for (全ての変数v2∈POINT[s, p]について繰り返す)

for (全ての集合X∈WSETについて繰り返す) {X=X-VEXP[s, v2];}

if (Y∈WSET, a∈YなるYが存在するとき) Y=Y∪(b);

else

if (Y∈WSET, b∈YなるYが存在するとき) Y=Y∪(a);

else WSET=WSET∪{(a, b)};

else if (中間命令sが関数fの呼び出しであるとき)

for (全ての変数v3∈CHANGE[s, f]について繰り返す)

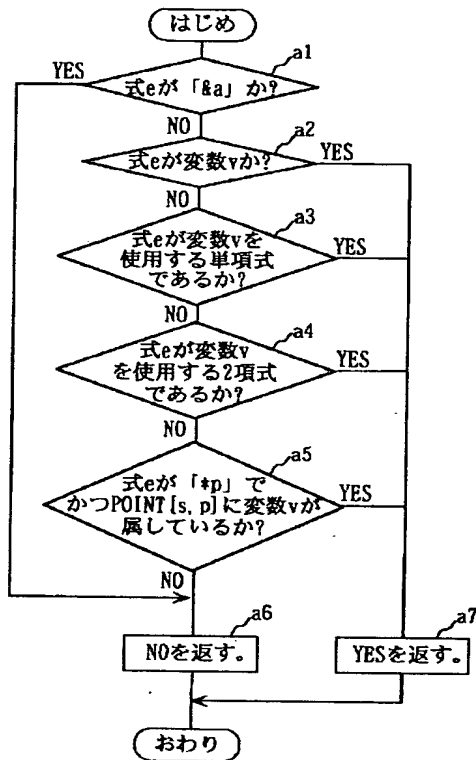
for (全ての集合X∈WSETについて繰り返す) {X=X-VEXP[v3];}

else if (中間命令sが条件分岐「if a cmp b goto label」

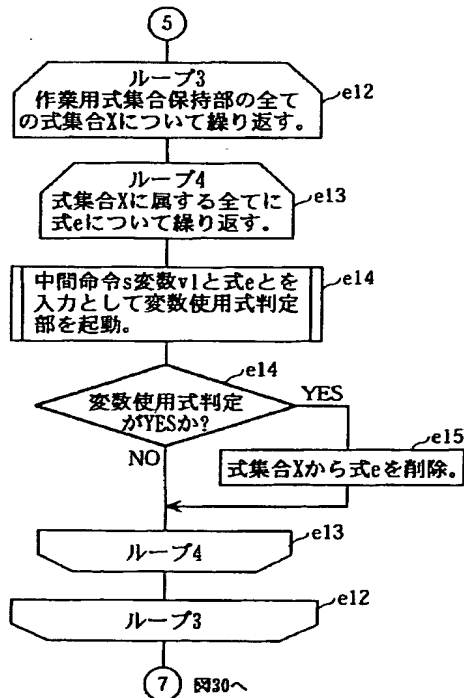
(cmp:関係演算子, label:分岐先ラベル)であるとき)

if (「a1 cmp b1」が恒真または恒偽であるa1, b1が存在するとき、但し、ここでa1は、
aそのものまたは、X1∈WSET, a∈X1において、a1∈X1なるものである。同様にb1は、
bそのものまたは、Y1∈WSET, b∈Y1において、b1∈Y1なるものである。)「a1 cmp b1」が恒真のときは、「a1 cmp b1」を「(1)」と
置き換え、恒偽のときは、「a1 cmp b1」を「(0)」と置き
換える。

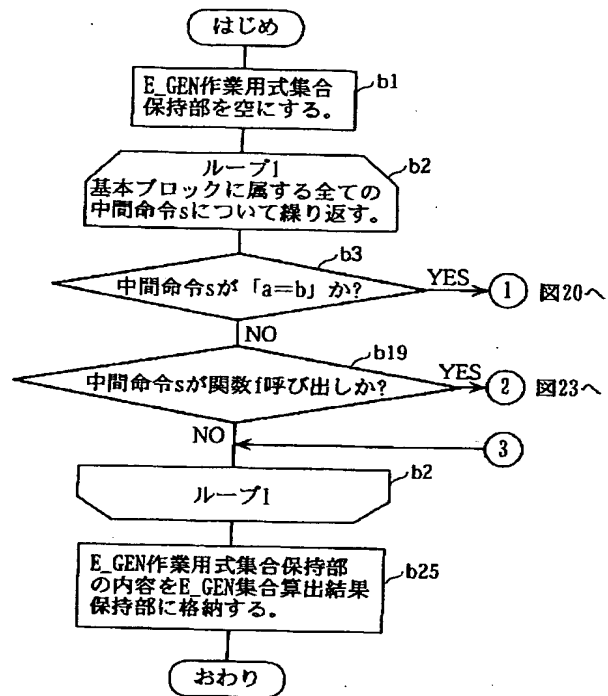
【 図18 】



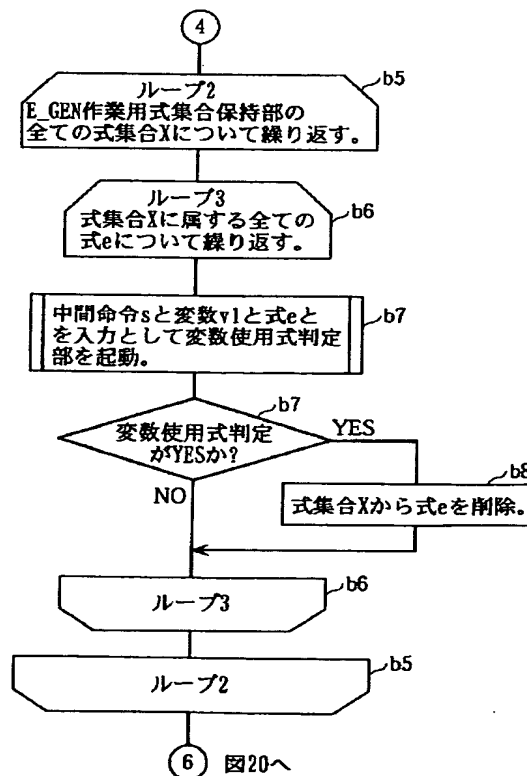
【 図31 】



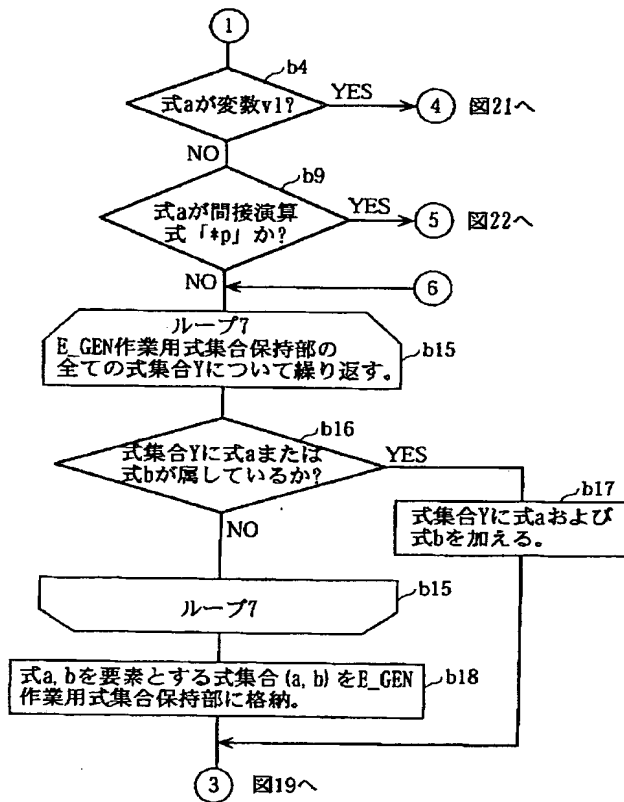
【 図19 】



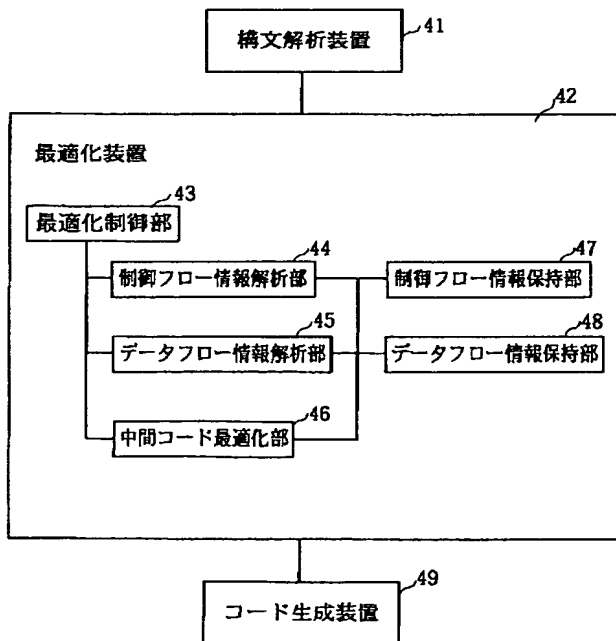
【 図21 】



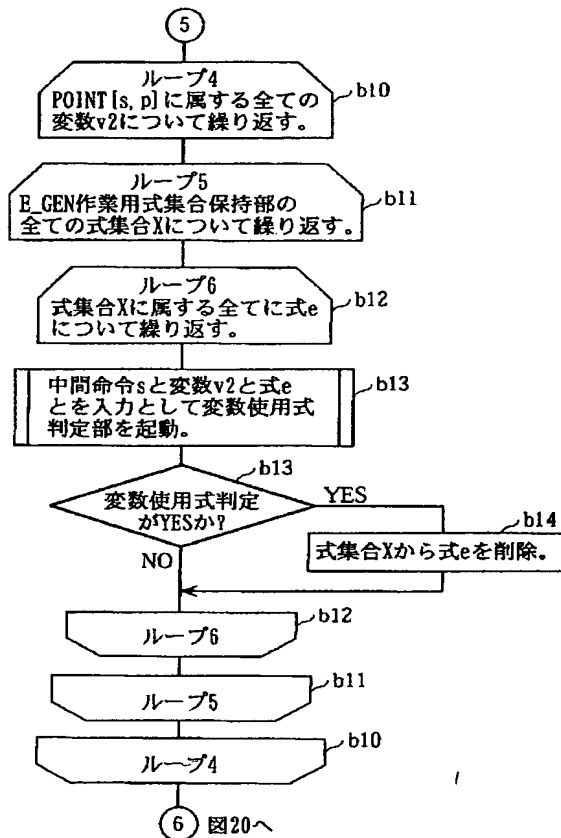
【 図20 】



【 図36 】



【 図22 】



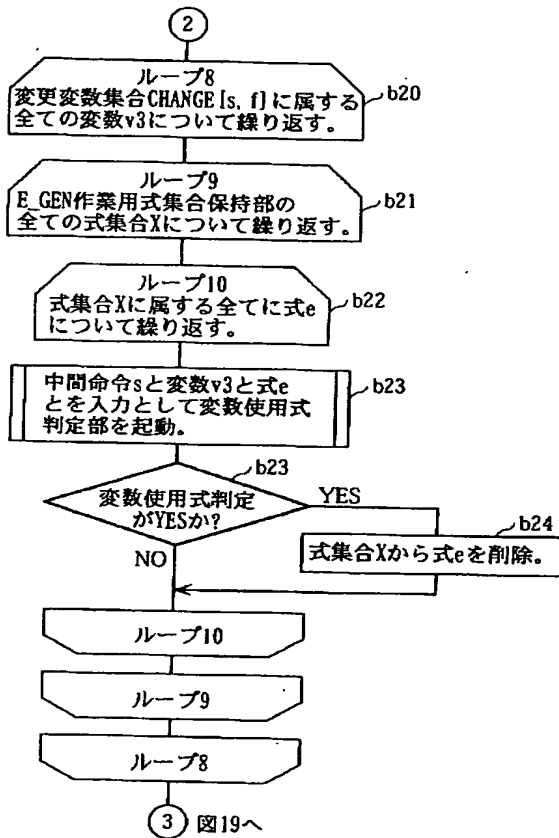
【 図37 】

[到達する定義の繰り返しアルゴリズム]

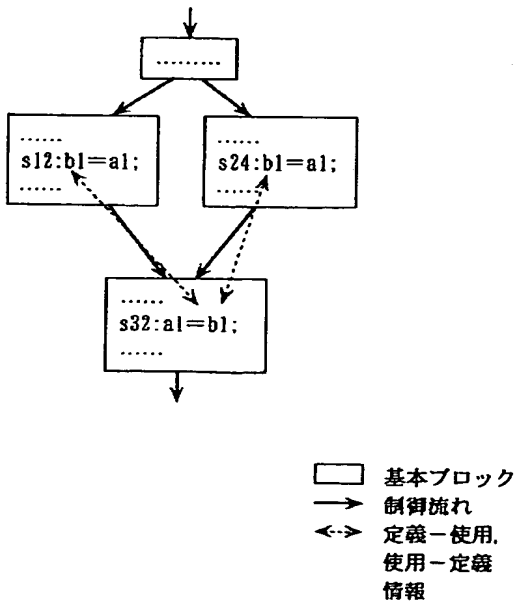
入力: 制御フロー情報と各基本ブロックBについてのGEN集合とKILL集合
出力: 到達する定義のIN集合、OUT集合

方法:
for (全ての基本ブロックBについて繰り返す) do OUT[B] = GEN[B];
repeat
 changed = false;
 for (全ての基本ブロックBについて繰り返す) |
 IN[B] = $\bigcup OUT[B']$;
 $B' \in pred(B)$
 old_OUT = OUT[B];
 OUT[B] = GEN[B] $\cup (IN[B] - KILL[B])$
 if (old_OUT \neq OUT[B]) changed = true;
 |
until changed == false;

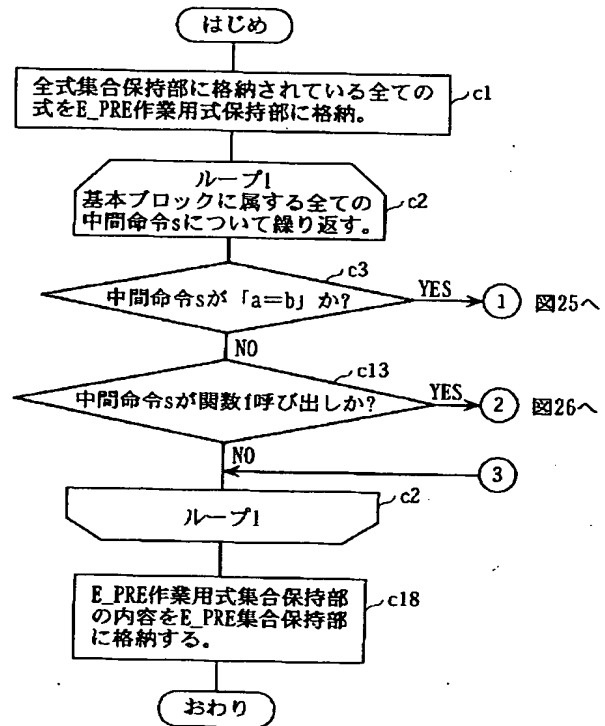
【 図23 】



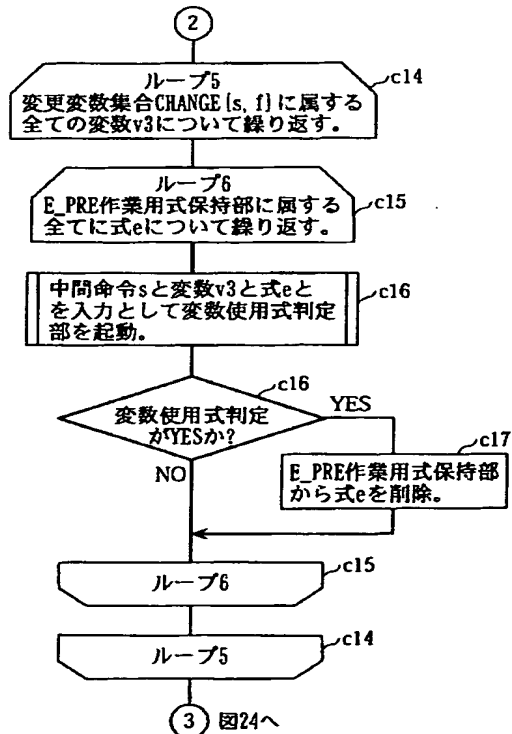
【 図38 】



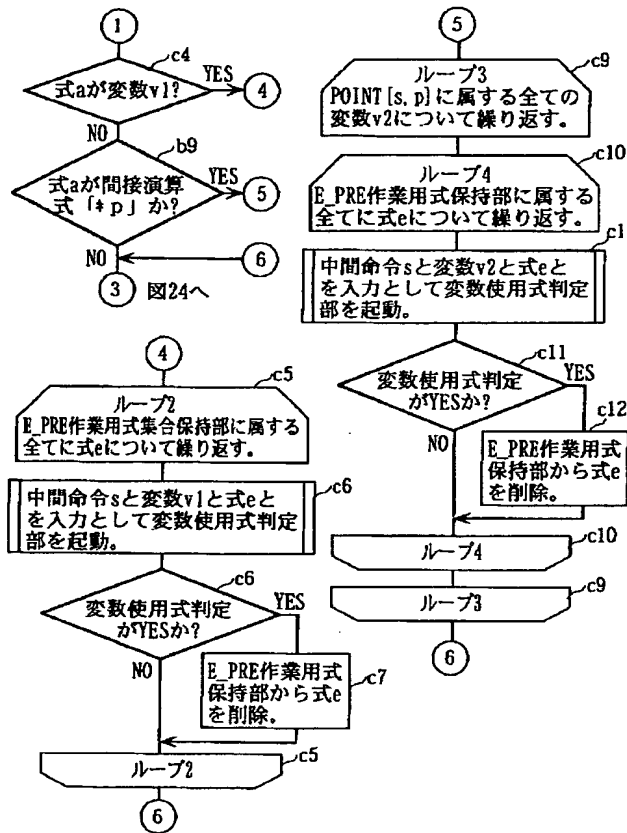
【 図24 】



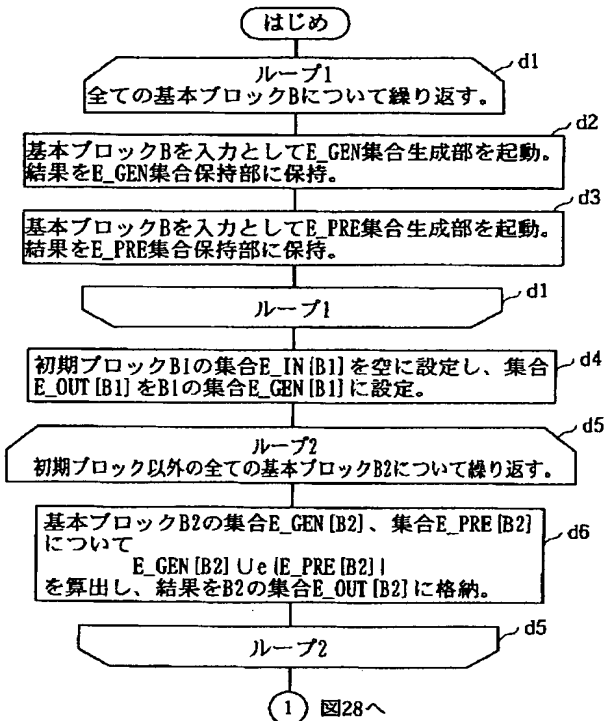
【 図26 】



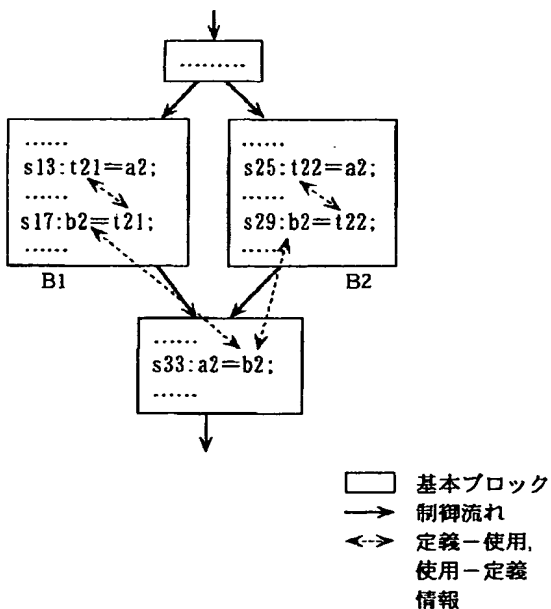
【 図25 】



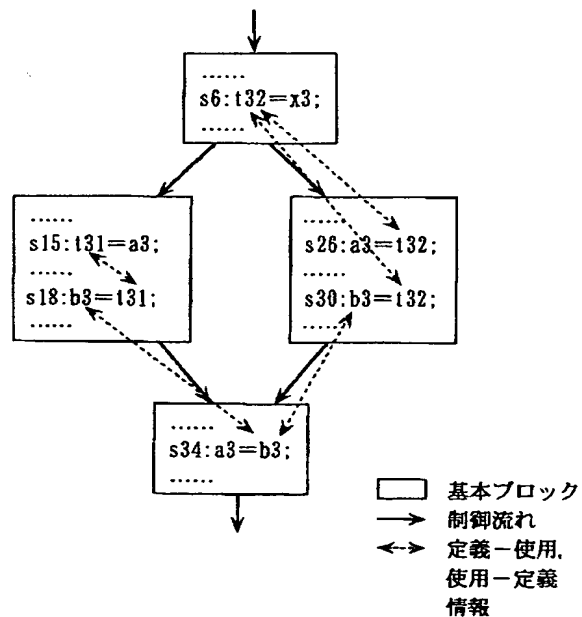
【 図27 】



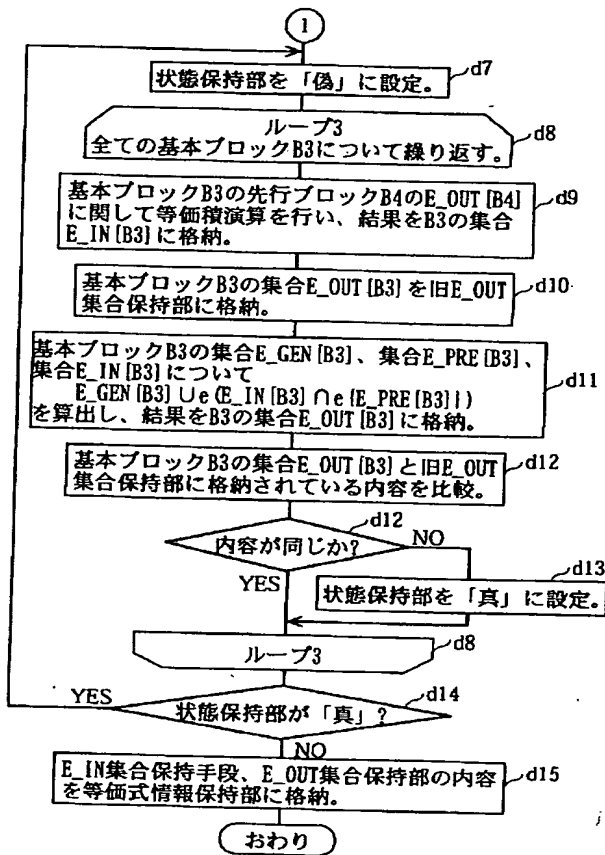
【 図39 】



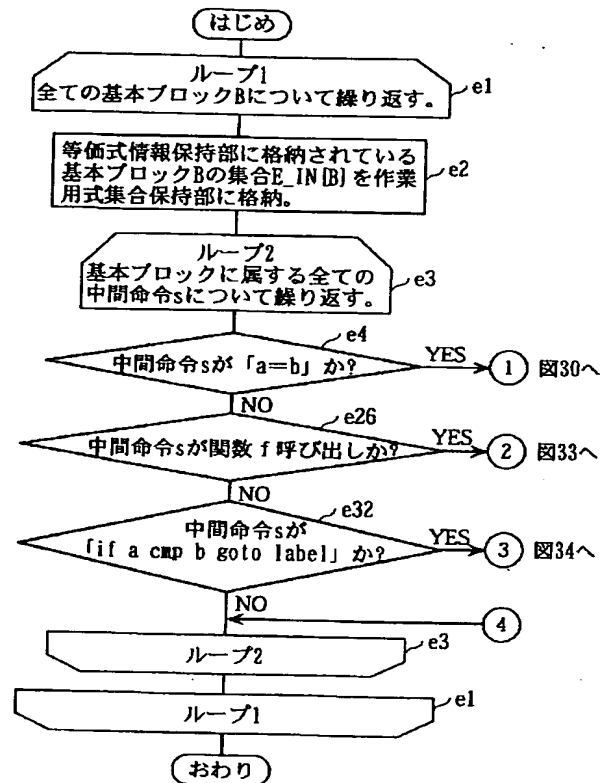
【 図40 】



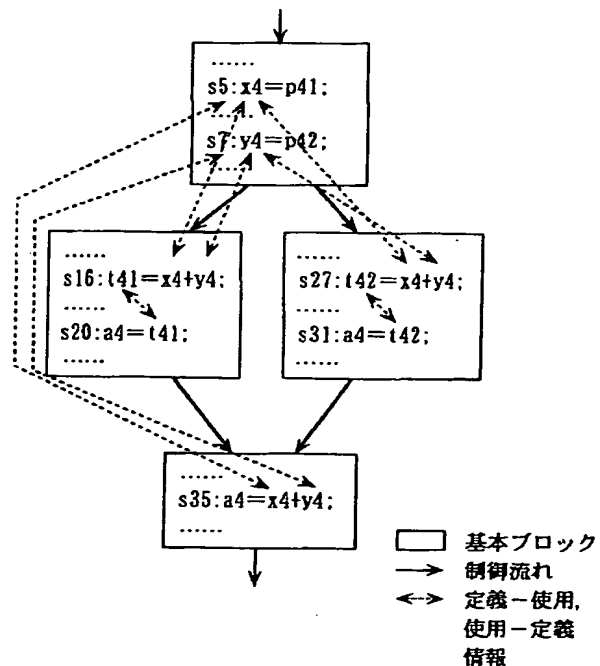
【 図28 】



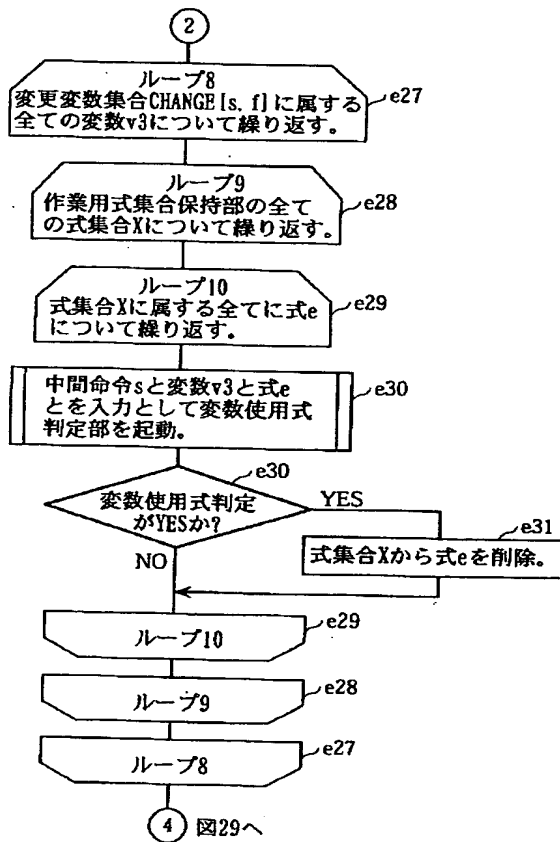
【 図29 】



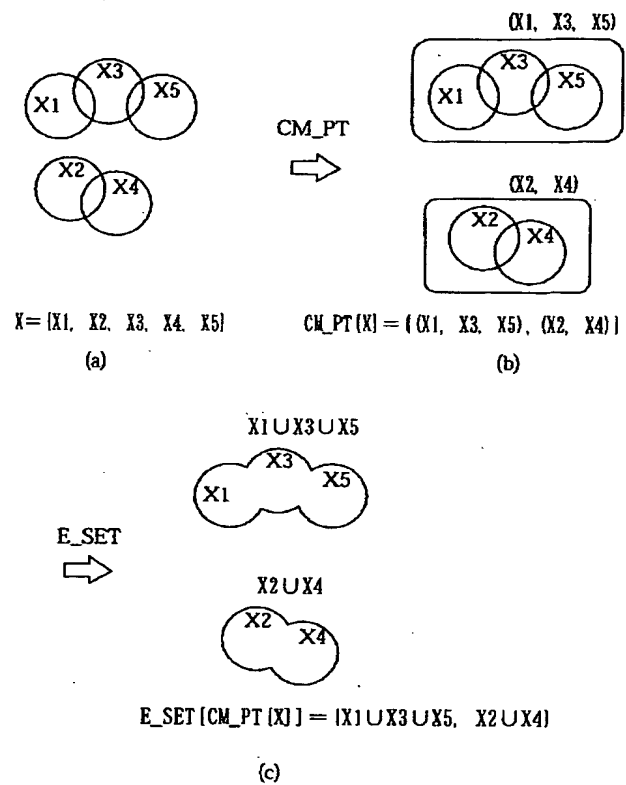
【 図41 】



【 図33 】



【 図35 】



【 手続補正書】

【 提出日】平成10年11月11日

【 手続補正1 】

【 補正対象書類名】図面

【 補正対象項目名】図16

【 補正方法】変更

【 補正内容】

【 図16 】

[等価式集合の計算アルゴリズム]

入力: 制御フロー情報と各基本ブロックBについてのE_GEN[B] とE_PRE[B]

出力: E_OUT[B], E_IN[B]

方法:

for (全ての基本ブロックBについて繰り返す) {

 E_GEN[B] を計算する。

 E_PRE[B] を計算する。

}

E_IN[B1] = ϕ ; /*B1: 初期ブロック*/

E_OUT[B1] = E_GEN[B1];

for (B1以外の全ての基本ブロックBについて繰り返す) {

 E_OUT[B] = E_GEN[B] \cup e {E_PRE[B]}; ... (1)

}

repeat

 changed = false;

 for (B1以外の全ての基本ブロックBについて繰り返す) {

 E_IN[B] = \cap e E_OUT[B'];

 B' \in pred[B]

 old_E_OUT = E_OUT[B];

 E_OUT[B] = E_GEN[B] \cup e {E_IN[B] \cap e {E_PRE[B]}};

 if (old_E_OUT \neq E_OUT[B]) changed = true;

 }

until changed == false;

【 手続補正2 】

【 補正対象書類名 】 図面

【 補正対象項目名 】 図1 7

【 補正方法 】 変更

【 補正内容 】

【 図1 7 】

[等価式置換アルゴリズム]

入力:基本ブロックB

出力:等価式置換を行なった基本ブロックB

方法:

WSET=E₁∪N(B):

for (Bの先頭の中間命令から終りの中間命令まで、順に中間命令sを取り出し以下を繰り返す。)

if (中間命令sが「a=b」であるとき)

if (a∈X, b∈XなるX∈WSETが存在するとき) (中間命令sを削除する。)

else

if (bが2項演算であるとき、つまりbは「b₁ op b₂」であるとき)if (「b₁ op b₂」が「定数c₁ op 定数c₂」に置き換え可能なき、但し、ここで、
c₁はb₁そのものまたは、X₁∈WSET, b₁∈X₁において、c₁∈X₁なるものである。同様に、c₂はb₂そのものまたは、X₂∈WSET, b₂∈X₂において、
c₂∈X₂なるものである。)「定数c₁ op 定数c₂」の計算結果を改めて中間命令sの式bとする。

|

else

if (bが単項演算であるとき、つまりbは「op b₁」であるとき)if (「op b₁」が「op 定数c₁」に置き換え可能なき、但し、ここで、c₁はb₁
そのものまたは、X∈WSETにおいてb₁∈Xかつ定数c₁∈Xなる、定数c₁
が存在するとき)「op 定数c₁」の計算結果を改めて中間命令sの式bとする。

|

if (aが変数v₁のとき)for (全てのX∈WSETについて繰り返す) (X=X-VEXP[s, v₁];)

|

else

if (aが間接演算式*_pであるとき)for (全ての変数v₂∈POINT[s, p]について繰り返す)for (全ての集合X∈WSETについて繰り返す) (X=X-VEXP[s, v₂];)

|

if (Y∈WSET, a∈YなるYが存在するとき) Y=Y∪(b);

else

if (Y∈WSET, b∈YなるYが存在するとき) Y=Y∪(a);

else WSET=WSET∪{(a, b)};

else if (中間命令sが関数fの呼び出しであるとき)

for (全ての変数v₃∈CHANGE[s, f]について繰り返す)for (全ての集合X∈WSETについて繰り返す) (X=X-VEXP[s, v₃];)

|

else if (中間命令sが条件分岐「if a cmp b goto label」

(cmp:関係演算子, label:分岐先ラベル)であるとき)

if (「a cmp b」が恒真または恒偽であるa₁, b₁が存在するとき、但し、ここでa₁は、
aそのものまたは、X₁∈WSET, a∈X₁において、a₁∈X₁なるものである。同様にb₁は、
bそのものまたは、Y₁∈WSET, b∈Y₁において、b₁∈Y₁なるものである。)「a cmp b」が恒真のときは、「a cmp b」を「(1)」と
置き換え、恒偽のときは、「a cmp b」を「(0)」と置き
換える。

|

|

【 手続補正3 】

【 補正対象書類名】明細書

【 補正対象項目名】0007

【 補正方法】変更

【 補正内容】

【 0007 】尚『プログラムの冗長性』には様々な解釈があるが、以降の説明における『プログラムの冗長性』は、高級言語或は中間言語にて記述されたプログラムにおいて、最終的に得られる機械語プログラムのコードサイズや実行時間を増大させる全ての要因を意味することとする。従来の最適化装置について述べる前に、コンパイラの構成について説明する。説明に先立ち、本発明で引用する文献を上げる。

[1] A.V.Aho, R.Sethi, J.D.Ullman: "Compilers, Principles, Techniques, and Tools", Addison Wesley Publishing Company Inc., 1986

(邦訳) 原田賢一: "コンパイラ I, II", サイエンス社, 1990

[2] Hans Zima: "Supercompilers for Parallel and Vector Computers", Addison Wesley Publishing Company Inc., 1991

(邦訳) 村岡洋一: "スーパーコンパイラ", オーム社, 1995

[3] 佐々政孝: "プログラミング言語処理系", 岩波書店, 1989

【 手続補正4 】

【 補正対象書類名】明細書

【 補正対象項目名】0044

【 補正方法】変更

【 補正内容】

【 0 0 4 4 】また、図4 1 の中間命令 s5において、変数 x4と変数 p41との等価関係が生成し、この関係が中間命令 s35の直前でも壊れず成立していることであるので、このことを変数 xと変数 p1の等価な関係が、中間命令 s35の直前に「到達する」ということにする。

(第1.2章 本実施形態で引用する数学的な記述および用語) 以下では、集合論における一般的な用法である記号を用いる。すなわち、集合の要素であることを「 \in 」で、集合の和を「 \cup 」、集合の共通部分を「 \cap 」、集合の差を「 $-$ 」、集合同士が等しいかまたは一方に含まれることを「 \subseteq 」で表現する。また空集合を「 ϕ 」と表現する。また共通部分が空集合である集合同士は「互いに素である」という。また、全称記号を「 \forall 」、存在記号を「 \exists 」で表現する。

{ 数式1 5 }

A, B, C \in PR(E) に対して、

$$\forall w_i \in (A \cap_e C) \cup_e (B \cap_e C),$$

$$\exists x_j \in (A \cup_e B) \cap_e C,$$

$$w_i \subseteq x_j$$

...(a)

(全ての $w_i \in (A \cap_e C) \cup_e (A \cap_e C)$ に対して、

$x_j \in (A \cup_e B) \cap_e C$ が存在し、 $w_i \subseteq x_j$ である。)

[証明]まず、 U_e の定義より、

(1) $\forall A_k \in A, \exists x_l \in A \cup_e B, A_k \subseteq x_l$ であり、

(2) $\forall B_m \in B, \exists x_n \in A \cup_e B, B_m \subseteq x_n$ である。

(3) (1)より、 $\forall C_p \in C$ について、

$$A_k \cap C_p \subseteq x_l \cap C_p$$

ここで、 $(A_k \cap C_p) \in (A \cap_e C)$ 、 $(x_l \cap C_p) \in (A \cup_e B) \cap_e C$ であり、 A_k, C_p は任意であるから、

$\forall A_k \in (A \cap_e C), \exists x_j \in (A \cup_e B) \cap_e C, A_k \subseteq x_j$ である。

(4) (3)と同様に(2)より、 $\forall C_q \in C$ について、

$$(B_m \cap C_q) \subseteq (x_n \cap C_q)$$

ここで、 $(B_m \cap C_q) \in (B \cap_e C)$ 、 $(x_n \cap C_q) \in (A \cup_e B) \cap_e C$ であり、 B_k, C_q は任意であるから、

$\forall B_k \in (B \cap_e C), \exists x_k \in (A \cup_e B) \cap_e C, B_k \subseteq x_k$ である。

(5) (3) (4)より、

$$\forall E_i \in ((A \cap_e C) \cup (B \cap_e C)), \exists y_m \in (A \cup_e B) \cap_e C, E_i \subseteq y_m$$

である。

(6)演算 U_e の定義より、

$w_i \in (A \cap_e C) \cup_e (B \cap_e C)$ に対して、

分割 $P_i = [E_k, E_{k+1}, \dots, E_n]$

($P_i \in \text{COMP}[(A \cap_e C) \cup (B \cap_e C)]$),

$E_i \in ((A \cap_e C) \cup (B \cap_e C))$ ($k=1..n$))

が存在し、 $w_i = U E_j$

$j=k..n$

である。

・式の全体集合 E

中間命令のプログラムに現れる条件式以外の式の全体集合を E とする。例えば、図3に記載されている式が全てであれば、式の全体集合 E は図5 (a)となる。

【 手続補正5 】

【 補正対象書類名】明細書

【 補正対象項目名】0 0 5 6

【 補正方法】変更

【 補正内容】

【 0 0 5 6 】以上により、文献 [3] p81の意味において、集合 PR(E) は演算 \cap_e おいてセミラテスである。

・演算子 U_e への演算子 \cap_e の分配

ここでは、後述の「関数 $f(x) = E \# \text{GEN}[B] \cup_e (x \cap_e [E \# \text{PRE}[B]])$ の単調性の証明」で使用する演算 U_e に演算 \cap_e を分配したときの次の数式1 5 の命題について証明する。

(7) (6)において、 P_i が一つの要素しかもたないとき、つまり $P_i = [E_k]$ であるときは、 $w_i = E_k$ となり (5)より、 $w_i \subseteq y_m$ となる $y_m \in ((A \cup_e B) \cap_e C)$ が存在する。

(8) \cap_e 関係が推移的であることから、(6)において P_i が複数の要素をもつときは、任意の $E_l, E_n \in P_i$ について、

$$E_l \cap E_n \neq \phi$$

.....

$$E_{t-1} \cap E_t \neq \phi$$

$$E_t \cap E_{t+1} \neq \phi$$

.....

$$E_{n-1} \cap E_n \neq \phi$$

なる、 $E_t \in P_i$ ($t=1..n$) が存在する。

(9) ここで、(5)より、 $E_t \subseteq Y_t, E_{t+1} \subseteq Y_{t+1}$ ($Y_t, Y_{t+1} \in (A \cup_e B) \cap_e C$) が存在し、かつ、 $E_t \cap E_{t+1} \neq \phi$ であるから、共通項 $Y_c = E_t \cap E_{t+1}$ は、 Y_t, Y_{t+1} 両方に含まれる。つまり、

$$Y_c \subseteq Y_t \text{ かつ } Y_c \subseteq Y_{t+1}$$

ところで、 $(A \cup_e B) \cap_e C \in \text{PR}(E)$ であり要素間は互いに素であるから、 $Y_t = Y_{t+1}$ となる。

(10) E_l, E_n は P_i の任意の要素であるので、結局、 P_i の全要素が Y_t に含まれ P_i の全要素の集合和である w_i は、 Y_t に含まれる。つまり、 w_i に対して、 $w_i \subseteq Y_t$ なる $Y_t \in ((A \cup_e B) \cap_e C)$ が存在する。

(11)結論

(7)と(10)より、(a) が証明された。(証明終り)

・変数使用式集合 (VEXP[s, v])

ある特定の中間命令 s1において変数 v の値が更新されると

き影響を受ける式の集合を、中間命令 s における変数 v の変数使用式集合と呼び、 $VEXP(s, v)$ と表現する。形式的には次の数式 16 に示すような式の集合である。

{ 数式 16 }

$VEXP(s, v) = \{e \mid e \text{ は } E \text{ に属する次の形の式である。但し、} \alpha \text{ はアドレス演算子} \& \text{ 以外の任意の演算子である。}\}$

【 手続補正 6 】

【 補正対象書類名 】 明細書

【 補正対象項目名 】 0075

【 補正方法 】 変更

【 補正内容 】

【 0075 】 本フローチャートにおいて第1 フェーズに相当するのは、ステップ d 2 における $E\#GEN$ 集合生成部 25 による $E\#GEN[B]$ の生成と、ステップ d 3 における $E\#PRE$ 集合生成部 27 による $E\#PRE[B]$ の生成とを全ての基本ブロック B について繰り返すループ 1 である。フローチャートにおいて第2 フェーズに相当するのは、初期化ブロック $B1$ の $E\#OUT$ 集合及び $E\#IN$ 集合の初期化を行うステップ d 4 と、等価和演算 $E\#GEN[B2] \cup E\#PRE[B2]$ を行うステップ d 6 を初期ブロック以外の全ての基本ブロック

$B2$ について繰り返すループ 2 である。

【 手続補正 7 】

【 補正対象書類名 】 明細書

【 補正対象項目名 】 0124

【 補正方法 】 変更

【 補正内容 】

【 0124 】 ステップ d 5 では、初期ブロック $B1$ 以外の基本ブロックを順次取り出し、取り出した基本ブロック $B2$ についてステップ d 6 の処理を実行する。初期基本ブロック $B1$ 以外の全ての基本ブロックについて処理を終了したらステップ d 7 へ進む。ステップ d 6 では、まず、全式集合保持部 23 に保持されている全ての式集合と、ステップ d 3 で算出される $E\#PRE$ 集合保持部 31 に保持されている $B2$ の $E\#PRE$ 集合に関して等価和演算を行なう。さらに演算結果と、ステップ d 2 で算出される $E\#GEN$ 集合保持部 30 に保持されている $B2$ の $E\#GEN$ 集合に関して、等価和演算を行ない、演算結果を $E\#OUT$ 集合保持部 33 に保持される、基本ブロック $B2$ の $E\#OUT$ 集合に保持する。ステップ d 5 へ進み次の基本ブロックを取り出す。

フロントページの続き

(72) 発明者 佐山 旬子

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)